UNIVERSIDAD DE GRANADA

Escuela Técnica Superior De Ingeniería Informática


**Departamento de Ciencias de la Computación e Inteligencia Artificial**

# MODELOS BAYESIANOS PARA LA CLASIFICACIÓN SUPERVISADA. APLICACIONES AL ANÁLISIS DE DATOS DE EXPRESIÓN GENÉTICA

TESIS DOCTORAL

**Francisco Javier García Castellano**

Granada 2009

Editor: Editorial de la Universidad de Granada
Autor: Francisco Javier García Castellano
D.L.: Gr. 157-2010
ISBN: 978-84-692-8389-9

# MODELOS BAYESIANOS PARA LA CLASIFICACIÓN SUPERVISADA. APLICACIONES AL ANÁLISIS DE DATOS DE EXPRESIÓN GENÉTICA

TESIS DOCTORAL

**Francisco Javier García Castellano**

DIRECTORES

**Serafín Moral Callejón**
**Andrés Cano Utrera**
**Luis Miguel de Campos Ibañez**

Granada 2009

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN E
INTELIGENCIA ARTIFICIAL
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

La memoria titulada

**Modelos bayesianos para la clasificación supervisada. Aplicaciones al análisis de datos de expresión genética**

que presenta D. Francisco Javier García Castellano, para optar al grado de **Doctor en Informática**, ha sido realizada en el *Departamento de Ciencias de la Computación e Inteligencia Artificial* de la *Universidad de Granada*, bajo la dirección del Dr. D. Serafín Moral Callejón, el Dr. D. Andrés Cano Utrera y el Dr. D. Luis Miguel de Campos Ibañez.

Granada, a de 2009

Fdo:D. Francisco Javier García Castellano

Fdo: D. Serafín Moral Callejón

Fdo: D. Andrés Cano Utrera

Fdo: D. Luis Miguel de Campos Ibañez

A mis padres, nadie se lo merece más que ellos.

# Agradecimientos

*"If I have seen further, it is by standing upon the shoulders of giants"*[1], es una frase escrita por Isaac Newton en una carta a Robert Hooke, hacia el año 1675 y que define exactamente el agradecimiento que quiero expresar aquí.

Quisiera empezar por los hombros que más arriba me han levantado en la realización de este trabajo, pues he tenido la suerte extraordinaria de contar con los doctores Don Serafín Moral Callejón, Don Andrés Cano Utrera y Don Luis Miguel de Campos Ibáñez, como directores de esta memoria. Gracias por vuestro apoyo constante, consejos, esfuerzo, tiempo, estímulo y, sobre todo, esa paciencia infinita que habéis demostrado conmigo. Ahora entiendo porqué existe este apartado de agradecimiento, es por personas como vosotros.

También quiero dar las gracias a los miembros del grupo de investigación *"Tratamiento de la Incertidumbre en Inteligencia Artificial"*, al cual pertenezco, por su apoyo y ánimos. Pero quisiera agradecérselo, en especial, a Silvia Acid por ayudarme con el motor de arranque y a Manolo Gómez por los tirones de orejas; no os pega el látigo, sois demasiado buenos. Tampoco puedo dejar de mencionar a Joaquín Abellán y a Andrés Masegosa, por las numerosas veces que me han prestado su ayuda.

Mi agradecimiento también va a los integrantes del proyecto Elvira, de todos he aprendido mucho. Pero quiero destacar, los dos meses que pasé en la Universidad del País Vasco, donde Pedro Larrañaga e Iñaki Inza, no sé como, en tan poco tiempo consiguieron enseñarme tantas cosas y ponerme en la pista de los clasificadores bayesianos. También por lo que aprendí de ellos y

[1] *"Si he visto más lejos es porque estoy sentado sobre los hombros de gigantes".*

por su amistad, no puedo olvidarme de Rubén Armañanzas, Rosa Blanco, Guzmán Santafé, José Luis Flores, Aritz Pérez y al resto de becarios que por entonces pululaba el *"Intelligent Systems Group"*.

Desde lo personal, si hay hombros en los que siempre me he apoyado, incondicionalmente, son los de mis padres Maribel Castellano e Ignacio García. Lo bueno que hay en mi, es responsabilidad casi exclusiva de ellos. A mi hermano Ignacio y a mi cuñada Conchi también quiero agradecerles su apoyo pero, sobre todo, haberme regalado a esos dos sobrinos, María y Miguel, que con su alegría y ganas de vivir son el mejor remedio para espantar mis males.

También quiero que mis amigos tengan un hueco: Bruno, Carlos, Víctor, Jesús, Lúcas y, como no, Yolanda. Gracias por preocuparos, por cuidarme y por ayudarme a desconectar.

Pero hay un personaje, que por su compañía, del que no me puedo olvidar: es el gigante más chico y más viejo, Coco, mi perrete. Mientras estaba enfrascado en la redacción de este trabajo, era el encargado de darme con la pata cuando estaba demasiado tiempo en frente del ordenador o de sacarme de paseo para que me diera un poco el aire.

Muchísimas gracias.

# Índice general

# Índice de figuras

# Índice de tablas

# Índice de algoritmos



# Introducción general.

En nuestra forma de relacionarnos con el mundo utilizamos, casi sin darnos cuenta, un razonamiento aproximado, es decir, no tenemos un conocimiento exacto de lo que pasa a nuestro alrededor, por ejemplo, sabemos que el litro de leche cuesta sobre un euro, que la gasolina sube un poco de forma frecuente o, es probable que mis padres estén comiendo solos en casa hacia el medio día. Nos movemos sin tener un conocimiento exacto de lo que pasa a nuestro alrededor y tampoco nos hace falta. Gracias a nuestra inteligencia, nos desenvolvemos razonablemente bien sin conocer todos los detalles.

Este desconocimiento o incertidumbre con el que nos relacionamos en el entorno que habitamos, sería muy beneficioso modelizarlo e implementarlo dentro de una máquina. ¿Beneficioso?. Imaginemos una máquina que nos diga de forma aproximada qué empresa del mercado bursátil va a aumentar más su valor, o una máquina que nos diga, con un pequeño margen de error, qué tiempo va a hacer hoy a cualquier hora del día y en cualquier parte del mundo o un coche que nos avise cuándo estamos cambiando involuntariamente de carril. Ahora imaginemos una máquina que mirando una muestra de un paciente le haga un diagnostico de forma casi inmediata y aproximada de las enfermedades o patologías de origen genético que tiene o pueda tener, es decir, algo del estilo "Usted, por sus genes tiene un 80 % de probabilidad de tener cáncer de pulmón sin fumar". ¿Se lo imagina? Y sin fumar.... Esta memoria pretende avanzar en esta dirección proponiendo métodos para razonar con conocimiento incierto. Quizás, en determinados casos, no sea aconsejable suplantar un experto con la decisión de una máquina pero sería absurdo despreciar esa información y no hacer uso de ella.

Lo primero que debe centrar nuestra atención es la forma de modelizar la incertidumbre; algo que no es nuevo, ya que los investigadores que trabajan en el área de la Inteligencia Artificial (IA, en lo sucesivo) llevan bastantes

años dedicando su esfuerzo a estudiar su modelización y utilización. De entre la gran cantidad de métodos que existen para representar la información disponible, la Teoría de la Probabilidad es quizás la más clásica y la más conocida.

Nosotros nos vamos a centrar en la Teoría de la Probabilidad desde la perspectiva bayesiana, donde la probabilidad de que ocurra un suceso representa el grado subjetivo de creencia que tiene un individuo sobre la realización del mencionado suceso, y gracias al teorema de Bayes y al teorema de la probabilidad total es posible actualizar la probabilidad cuando se obtiene un aporte de una nueva información. La nueva perspectiva bayesiana nos permite modelizar los conceptos de relevancia o independencia, nociones que las personas utilizamos de forma habitual en nuestro razonamiento.

En los primeros trabajos que utilizaron la probabilidad para tratar la incertidumbre se encontraron problemas debido a la necesidad de asumir algunas hipótesis de independencia poco realistas que se usaban para reducir el número de parámetros a calcular y a la imposibilidad de una asignación o estimación precisa de éstos. Por este motivo, al comienzo, se dejó a un lado el uso de la probabilidad para manejar la incertidumbre. No obstante, con la aparición de las redes bayesianas la probabilidad ha sido aceptada como la forma más intuitiva y eficaz para medir la incertidumbre.

El que dos hechos $X$ e $Y$ sean independientes nos viene a decir que el conocer que la primera variable $X$ toma un determinado valor, no aporta ninguna información acerca del valor que puede tomar la segunda variable $Y$ y viceversa. Por ejemplo, supongamos que tenemos dos dados: si los lanzo y en el primero me sale un 2 y en el segundo un 5, el primer valor, el 2, no condiciona para nada el valor del segundo dado, el 5, es decir, el resultado del segundo dado es independiente del primero y al revés. La idea es modelizar el conocimiento de aquellas variables que son independientes para no usar información irrelevante y que, además, la información que si es relevante sea fácilmente accesible.

Las redes bayesianas están compuestas de una parte gráfica y una parte cuantitativa, más concretamente, un grafo dirigido acíclico y una colección de parámetros numéricos, normalmente tablas de probabilidad condicionada.

La parte gráfica nos va a permitir representar de forma explícita relaciones de dependencia e independencia entre variables. De esta forma, en el grafo los nodos representan a las variables y la relación de dependencia entre dos variables se representa mediante la existencia de un camino o un arco entre

ellas. Ahondando en esta idea, en la red bayesiana, si dos variables $X$ e $Y$ están conectadas por un arco $X \rightarrow Y$ sabemos que las dos variables están relacionadas. Simplemente mirando al grafo y viendo las conexiones entre las variables podemos saber si son o no independientes entre ellas.

La parte cuantitativa de nuestra red es la información numérica necesaria para determinar una distribución conjunta teniendo en cuenta las independencias representadas en la parte cualitativa. Nos va decir en qué medida nos creemos las relaciones de dependencia entre las variables, permitiéndonos así representar la incertidumbre. De esta forma, si conocemos un conjunto determinado de hechos podemos calcular la probabilidad de que ocurra lo que no conocemos. Este tipo de conocimiento se proporcionará mediante un conjunto de distribuciones de probabilidades condicionadas.

Resumiendo, podemos decir que una red bayesiana es una representación gráfica de una distribución de probabilidad conjunta, gracias a la cual normalmente el número de parámetros a estimar se reduce de forma sustancial.

Una red bayesiana puede ser construida manualmente utilizando un experto en el proceso y representando su conocimiento e incertidumbre en nuestro modelo, pero esto no siempre es posible por muchos motivos. Por ejemplo que no haya ningún experto en la materia, o que no podamos disponer de él, que no consigamos modelizar correctamente el conocimiento del experto o que el problema tenga tantas variables que el proceso sea tan complejo que sea inabordable en la práctica. Por estos motivos y muchos otros, se utilizan métodos de aprendizaje automático. De esta forma, un programa a partir de unos datos de ejemplo obtendrá un modelo que permite modelizar el conocimiento que hay en los mismos. El aprendizaje automático es fundamental en la IA, ya que si una máquina no es capaz de aprender o adaptarse por si sola ¿la podríamos considerar inteligente?.

El aprendizaje automático de las redes bayesianas se dividirá lógicamente en dos partes: el aprendizaje de la estructura gráfica y el aprendizaje de los parámetros numéricos correspondientes. Ha habido una gran cantidad de trabajos en el aprendizaje automático de redes bayesianas a partir de datos y, consecuentemente, hay una gran cantidad de algoritmos de aprendizaje basados en diferentes metodologías.

Las redes bayesianas pueden utilizarse en tareas de clasificación supervisada que podemos considerar como un subcampo del aprendizaje automático. En clasificación supervisada un conjunto de variables a considerar será $\mathbf{V} = \mathbf{U} \cup \{C\}$. Donde $C$ es la variable distinguida, denominada variable a clasificar

o, simplemente, *clase* y las variables en $\mathbf{U}$ serán los *atributos* o *características* usados para predecir los valores de $C$. El objetivo es calcular la probabilidad posterior de la clase dada cualquier configuración de los atributos, $p(C|\mathbf{u})$. Por ejemplo, tenemos un conjunto de síntomas (atributos) y una variable (clase) cuyos valores pueden tomar las distintas enfermedades que se pueden asociar con dichos síntomas. La idea es construir un programa que a partir de los síntomas, nos diga aproximadamente la enfermedad que tenemos, si la hubiere.

Las redes bayesianas se han aplicado, con resultados satisfactorios, en la solución de problemas en gran cantidad de dominios. Entre los que podemos destacar la medicina o la biología. Es por ello que las encontramos como una herramienta perfectamente válida para el tratamiento de datos de expresión genética. Pero veamos a qué nos referimos cuando hablamos de datos de expresión genética.

Los genes son secuencias de ADN que codifican la información necesaria para sintetizar las proteínas, las cuales son esenciales ya que se encargan de llevar a cabo prácticamente todas las funciones de la célula. En la transcripción de un gen, éste se activa para dar lugar a la proteína que codifica, en ese caso diremos que ese gen se está expresando dentro de la célula. Los microarrays de ADN nos permiten estudiar miles de genes simultáneamente, viendo qué genes están expresados y cuales no, dada una muestra de un tejido. A los datos provenientes de los microarrays de ADN también se les llama datos de expresión genética.

Los microarrays de ADN (también llamados chips de ADN) están formados, generalmente, por un substrato de cristal o de nylon, donde densamente se ordenan, en forma de matriz, cadenas simples de oligonucleótidos. Si se pone el microarray en contacto con una mezcla de cadenas simples de ADN a identificar, éstas se hibridarán con su oligo complementario.

Al poder estudiar tantos genes a la vez, esta técnica se ha aplicado para un mayor entendimiento de varios procesos biológicos como son la identificación de genes implicados en complejos procesos humanos, como, por ejemplo, el cáncer y la búsqueda de nuevos fármacos o en el estudio de diferentes procesos en otros organismos. Téngase en cuenta que la disponibilidad de datos de expresión genética nos va a permitir comprender procesos celulares desconocidos, realizar diagnósticos y tratamientos de enfermedades y el diseño de fármacos con una función conocida a nivel genético.

Los datos que se generan con microarrays tienen una gran cantidad de

variables, una por cada gen que se está estudiando, y suele haber pocos casos debido a que la técnica de microarrays es bastante costosa. Por lo tanto, tenemos un pequeño número de casos con una gran cantidad de variables a analizar. Otro problema al que nos enfrentamos en este tipo de datos es la gran cantidad de *ruido* que poseen. Tengamos en cuenta que estamos trabajando a un nivel microscópico donde una simple mota de polvo es un *gran* problema.

Recientemente se han aplicado sistemas gráficos probabilísticos (redes bayesianas) a estos datos. Las redes bayesianas al ser herramientas flexibles y visualmente interpretables, nos permiten representar relaciones de dependencia condicional entre genes y su naturaleza probabilística les permite tratar el ruido. Representando los genes como variables de la red bayesiana vamos a poder representar interacciones entre genes.

Para finalizar esta breve introducción, quisiéramos mencionar el entorno Elvira, que es una biblioteca de clases escrita en el lenguaje de programación Java, con un entorno gráfico, dirigido al diseño y uso de aplicaciones basadas en sistemas gráficos probabilísticos, con importantes contribuciones de nuevos algoritmos de aprendizaje e inferencia. Por este motivo y muchos otros, se ha elegido esta plataforma para el desarrollo de los diferentes metodologías que en esta tesis se van a presentar.

# Objetivos.

Teniendo en cuenta las dificultades que presentan los datos de expresión genética, nos encontramos ante un problema que, como se verá, ha generado un gran número de trabajos y estudios. Por tanto, los distintos objetivos que nos planteamos en esta memoria son los siguientes:

- Estudio previo de los métodos gráficos probabilísticos como mecanismo de representación de la incertidumbre y como herramienta de aprendizaje automático. Así, estudiaremos las distintas técnicas existentes para el aprendizaje automático de estos modelos a partir de datos.

- Estudio previo del uso de los métodos gráficos probabilísticos en el análisis de datos de expresión genética. Estudiaremos los distintos trabajos realizados en este campo, observando la metodología utilizada por cada autor; de esta forma podremos analizar la viabilidad de la utilización de estos modelos para el tratamiento de datos provenientes

de microarrays de ADN y aprender de las virtudes y fallos que encontremos en investigaciones previas.

- Incorporar la utilización de conocimiento experto en el aprendizaje de los modelos. Es evidente que se hace casi imposible que un experto nos diseñe de forma manual una red bayesiana para un problema donde tenemos miles de variables (genes) como ocurre en los datos de expresión genética, y además tengamos en cuenta que gran parte de las interacciones entre genes son desconocidas. No obstante, obsérvese la utilidad de aportar el conocimiento biológico que se posee entre interacciones de algunos genes a la hora de realizar el aprendizaje de datos con tantas variables y tan poco número de casos. Puede ser útil incorporar el conocimiento de un experto siempre y cuando se disponga del mismo.

- Desarrollo de nuevos algoritmos de aprendizaje automático para tareas de clasificación supervisada que obtengan mejores resultados que los existentes en la literatura.

- Validación de todos los métodos propuestos en conjuntos de datos usados comúnmente en Inteligencia Artificial y en datos de expresión genética.

- Incorporación de todos los modelos implementados al entorno Elvira.

## Descripción por capítulos.

En el primer capítulo nos dedicamos a dar unas nociones básicas de teoría de la probabilidad para seguidamente definir un poco más a fondo los modelos gráficos probabilísticos y más concretamente las redes bayesianas. Se hará una revisión de las distintas técnicas de aprendizaje de redes bayesianas y se hará también inciso en el uso de las redes bayesianas como clasificadores. Se analizarán también los pasos necesarios que han de sufrir los datos -preprocesamiento de los datos- para ser utilizados con las redes bayesianas. Finalmente, se presentará el entorno Elvira. En este capítulo se tratará de cumplir el primer subobjetivo propuesto en el anterior apartado.

En el segundo capítulo se verá una pequeña introducción a la genética para después presentar la técnica de microarrays de ADN, técnica con la cual se obtienen los datos de expresión genética, se presentarán los problemas que

conllevan este tipo de datos y, sobre todo, en su utilización con modelos gráficos probabilísticos. Además se hará una revisión de los trabajos que hay en la literatura acerca del uso de las redes bayesianas y los clasificadores bayesianos en su aplicación al estudio de datos de expresión genética. De esta forma se llevará a cabo el segundo de los subobjetivos propuestos.

En el tercer capítulo se estudiará la forma de representar el conocimiento del experto a la construcción de los modelos gráficos probabilísticos ya que no tendría sentido volver a descubrir o a estudiar funciones de genes ya conocidas, es decir, aquí se presentarán métodos para que la red bayesiana utilice información de un experto de forma que la parte de aprendizaje automático sólo se encargue de estudiar aquellos datos donde hay incertidumbre. De esta forma se presentará primero una formalización para representar dicha información experta adecuándose a su utilización en algunos algoritmos de aprendizaje automático de redes bayesianas.

Verificando el objetivo de estudiar nuevos algoritmos de aprendizaje para tareas de clasificación supervisada, en los capítulos cuarto, quinto y sexto se presentarán distintos clasificadores. En el capítulo cuarto, se estudiará un nuevo árbol de clasificación que mejore a otros algoritmos clásicos, mediante una estimación bayesiana de las probabilidades utilizadas. Para superar la incapacidad de representar independencias asimétricas que poseen los clasificadores bayesianos y las redes bayesianas, en general, en el quinto capítulo se estudiará la utilización de multiredes bayesianas en tareas de clasificación. En el sexto capítulo y con la idea de superar las limitaciones estructurales de las que adolecen la mayoría de los clasificadores bayesianos actuales se estudiarán modelos sin restricciones estructurales.

En el séptimo capítulo se estudiará la aplicabilidad de los nuevos algoritmos propuestos en los anteriores capítulos en su utilización en el análisis y clasificación de datos de expresión genética, cumpliendo así, con el penúltimo objetivo propuesto. En este capítulo se estudiará la viabilidad del uso de los modelos propuestos, haciendo hincapié en el preprocesamiento de los datos y se compararán los resultados experimentales con otras propuestas existentes en la literatura.

Finalmente, tendremos un capítulo de conclusiones donde se estudiarán los resultados obtenidos con los distintos algoritmos propuestos y su aplica-

ción práctica. También se presentarán posibles trabajos futuros para continuar con la labor de investigación que aquí se presenta.

# Capítulo 1

# Modelos gráficos probabilísticos.

Podemos decir que la Inteligencia Artificial tiene como objetivo la creación de programas que realicen tareas que requieran un comportamiento inteligente.

En nuestra forma de relacionarnos con el mundo utilizamos, casi sin darnos cuenta, incertidumbre, es decir, no tenemos un conocimiento exacto de lo que pasa a nuestro alrededor. Por ejemplo, no nos acordamos literalmente lo que decía el noticiario del día anterior pero, si le prestamos un poco de atención, sabemos cuales fueron las noticias más relevantes del día. Por tanto, si queremos que un programa imite el comportamiento del ser humano, desenvolviéndose con problemas del mundo real, debería ser capaz de manejar la incertidumbre existente sobre el problema que está tratando.

Desde hace ya varios años la Inteligencia Artificial ha dedicado un considerable esfuerzo al tratamiento de la incertidumbre. De entre todos los métodos que se han propuesto, la Teoría de la Probabilidad es la más clásica y la más conocida, sobre todo la óptica bayesiana. Las redes bayesianas son una herramienta que ha demostrado su capacidad como modelo de representación del conocimiento con incertidumbre en Inteligencia Artificial, siendo capaces de adaptarse con éxito a gran número de aplicaciones prácticas.

Las redes bayesianas permiten representar el conocimiento de forma gráfica y compacta, usando los conceptos de probabilidad y causalidad o indepen-

dencia entre las variables de un problema, de forma muy parecida a cómo el ser humano representa el conocimiento. Pero además de ser fácilmente interpretables, tienen la capacidad de obtener explicaciones de la información que representan y se adaptan con facilidad ante la llegada de nueva información.

En este capítulo se hará una breve introducción de conceptos de probabilidad para luego poder explicar detalladamente lo que son las redes bayesianas como un tipo de modelo gráfico probabilístico y su situación actual.

## 1.1. Sistemas expertos probabilísticos.

Un **sistema experto** lo podemos definir como un *sistema informático que simula a los expertos humanos en un área de especialización determinada* [63]. Los sistemas expertos que tratan la información de una forma determinista son poco realistas, ya que el conocimiento humano es en su mayoría heurístico, es decir, aproximado. Si queremos construir un sistema experto lo tendremos que dotar de la capacidad para razonar de forma aproximada, o lo que viene a ser lo mismo con incertidumbre.

Los **sistemas expertos probabilísticos** utilizan la probabilidad como medida de incertidumbre en sus razonamientos. Sin embargo, en los primeros sistemas expertos se utilizaron factores de certeza [66] como medida para tratar la incertidumbre pero requerían mucha información y unos cálculos demasiado complejos para poder resolver problemas reales en los que interviniesen un gran número de variables.

Con la aparición de los primeros **modelos gráficos probabilísticos** [261, 263, 212] (entre los que destacamos las redes de Markov y las redes bayesianas) se comprobó que las dificultades en el uso de la probabilidad eran superables, de hecho, actualmente la probabilidad es la medida de incertidumbre más aceptada.

La idea básica que subyace en los modelos gráficos probabilísticos es codificar el conocimiento de forma que no sea necesario utilizar información irrelevante y, por tanto, al trabajar con una complejidad menor, disminuir la complejidad de cálculo. Lo que se hace es aprovechar las relaciones de dependencia e independencia entre las variables de un problema -codificadas de forma gráfica en el modelo-, antes de especificar y calcular los valores numéri-

cos de las probabilidades. Estas relaciones se representan mediante modelos gráficos, habitualmente grafos dirigidos acíclicos [356].

## 1.2. Probabilidad.

La definición clásica o frecuentista de la probabilidad nos dice que es la frecuencia relativa del número de veces que se lleva a cabo un experimento cuando el número de repeticiones tiende a infinito, es decir, la razón entre el número de veces que se obtiene una determinada salida y el número de veces total que se realiza el experimento [59]. Por ejemplo, si cogemos 100 veces una carta de una baraja y 5 de esas veces nos aparece un as, entonces la probabilidad de que al descubrir una carta nos salga un as, la calculamos mediante la división 5/100, es decir, $p(carta = as) = 0{,}05$.

Más formalmente, la probabilidad $p$ de aparición de un suceso $A$ de un total de $n$ casos posibles sería $p(A)$ y se define como la razón entre el número de ocurrencias $m$ en que dicho suceso es cierto y el número total de casos posibles $n$:

$$p(A) = m/n$$

Esta definición tiene el problema que las frecuencias sólo son exactas en el límite de infinitas repeticiones. Actualmente se considera la probabilidad de $A$ como la propensidad de $A$ [91].

En lugar de esta aproximación frecuentista también podemos definir la probabilidad desde un punto de vista subjetivo [218] como la creencia de un individuo concreto sobre el resultado de un determinado experimento.

Como ya se ha indicado, la probabilidad es una herramienta que nos va a permitir modelar nuestro conocimiento aproximado sobre un suceso. Por tanto, la probabilidad nos va a permitir tratar con fenómenos no deterministas, es decir, para representar nuestra incertidumbre sobre un fenómeno.

En 1933, el matemático ruso Andrei N. Kolmogorov estableció un conjunto de axiomas (*Axiomas de Kolmogorov* o *Axiomas de probabilidad*) [200]

que deben satisfacerse para que podamos determinar consistentemente la probabilidad sobre unos sucesos. Dichos axiomas son:

- *Primer Axioma*: la probabilidad de un suceso A es un número real no negativo, es decir:
$$p(A) \geq 0$$
- *Segundo Axioma*: la probabilidad del espacio muestral $U$ es 1:
$$p(U) = 1$$
- *Tercer Axioma*: si $A_1, A_2, \ldots, A_n$ son un conjunto de sucesos mutuamente excluyentes (de intersección vacía dos a dos), entonces la probabilidad de que al menos uno de estos sucesos ocurra, es la suma de las probabilidades individuales:
$$p(A_1 \cup A_2 \cup \ldots \cup A_n) = \sum_{i=1}^{n}(p(A_i))$$

De estos axiomas hay una serie de propiedades que se pueden deducir:

- *Normalización*: $p(\phi) = 0$
- *Monotonicidad*: si $A_1 \subseteq A_2 \subseteq U$ entonces $p(A_1) \leq p(A_2)$
- *Inclusión-Exclusión*: dado cualquier par de subconjuntos $A_1$ y $A_2$ de $S$, se cumple siempre la siguiente igualdad:
$$p(A_1 \cup A_2) = p(A_1) + p(A_2) - p(A_1 \cap A_2)$$
- Para cualquier suceso $p(A) \leq 1$.
- Como $A$ y su complementario $\overline{A}$ son dos sucesos disjuntos, es decir, $U = A \cup \overline{A}$, podemos deducir que $p(\overline{A}) = 1 - p(A)$.

En un *experimento aleatorio*[1] nos va a hacer falta cuantificar los efectos de modo que se asigne un número real a cada uno de los resultados posibles

[1] Un experimento aleatorio es aquel que bajo el mismo conjunto aparente de condiciones iniciales, puede presentar resultados diferentes, como por ejemplo, lanzar un dado.

del experimento. De este modo, definimos una variable aleatoria $X$ como una función que toma valores en un conjunto $\Omega$ de acuerdo con una medida de probabilidad.
Supongamos que lanzamos un dado. El dominio de la variable sería cada uno de los valores que puede tomar el dado, es decir, $\Omega = \{1, 2, 3, 4, 5, 6\}$. Diremos que una variable aleatoria es *discreta* si su dominio es un conjunto numerable, como en el anterior ejemplo, y diremos que una variable aleatoria es *continua* en caso contrario.

Sean $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$ un conjunto de variables aleatorias discretas, supongamos que cada variable $X_i$ toma valores de un conjunto finito de estados $\Omega_{X_i}$ (el dominio de $X_i$). Vamos a usar $x_i$ para denotar los valores concretos de $X_i$, $x_i \in \Omega_{X_i}$, por ejemplo, si $X_i$ es una variable binaria que representa si una persona tiene gripe o no, entonces $x_i$ puede ser *sí* o *no*. Obsérvese que las variables aleatorias las vamos a denotar con mayúsculas y que los valores que pueden tomar dichas variables los vamos a denotar con minúsculas. Asimismo usaremos negrita y mayúsculas para representar conjuntos de variables y minúscula y negrita para conjuntos de valores de un conjunto de variables.

Veamos ahora una serie de definiciones que nos serán de gran utilidad.

La **distribución de probabilidad** de una variable aleatoria $X$ es una función que asigna a cada evento definido sobre la variable aleatoria una probabilidad. La distribución de probabilidad describe el rango de valores de la variable aleatoria así como la probabilidad de que el valor de la variable aleatoria esté dentro de un subconjunto de dicho rango.

***Probabilidad Conjunta y probabilidad Marginal***: sea $p(x_1, x_2, \ldots, x_n)$ la distribución de *probabilidad conjunta* sobre $X_1, X_2, \ldots, X_n$, es decir

$$p(x_1, x_2, \ldots, x_n) = p(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n)$$

Entonces la distribución de *probabilidad marginal* sobre la i-ésima variable se obtiene mediante la siguiente fórmula:

$$p(x_i) = p(X_i = x_i) = \sum_{x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n} p(x_1, x_2, \ldots, x_n)$$

***Probabilidad Condicional***: sean $X$ e $Y$ dos variables que toman valores en $\Omega_X$ y $\Omega_Y$ tales que $p(y_j) > 0 \ \forall y_j \in \Omega_Y$. Entonces la *probabilidad condicional* de $X \ \forall x \in \Omega_X$ dado $Y = y$ viene dada por

$$p(X = x|Y = y) = p(x|y) = \frac{p(x, y)}{p(y)}$$

Por tanto, la distribución de probabilidad conjunta de $X$ e $Y$ puede obtenerse como:

$$p(x, y) = p(y)p(x|y)$$

***Independencia y Dependencia Probabilística***: las siguientes definiciones nos van a permitir establecer las relaciones de dependencia o independencia entre variables:

*Dependencia e Independencia marginal*: sean $\mathbf{X}$ e $\mathbf{Y}$ dos subconjuntos disjuntos del conjunto de variables aleatorias. Se dice que $\mathbf{X}$ es *marginalmente independiente* de $\mathbf{Y}$, o simplemente, $\mathbf{X}$ es independiente de $\mathbf{Y}$, si y solamente si para $\forall \mathbf{x} \in \Omega_{\mathbf{X}}$ y $\forall \mathbf{y} \in \Omega_{\mathbf{Y}}$ se verifica que

$$p(\mathbf{x}|\mathbf{y}) = p(\mathbf{x})$$

En otro caso, se dice que $\mathbf{X}$ es (marginalmente) dependiente de $\mathbf{Y}$.

*Dependencia e Independencia condicional*: sean $\mathbf{X}$,$\mathbf{Y}$ y $\mathbf{Z}$ tres conjuntos disjuntos de variables. Se dice que $\mathbf{X}$ es *condicionalmente independiente* de $\mathbf{Y}$ dado que conocemos $\mathbf{Z}$, si y solamente si para $\forall \mathbf{x} \in \Omega_{\mathbf{X}}$, $\forall \mathbf{y} \in \Omega_{\mathbf{Y}}$ y $\forall \mathbf{z} \in \Omega_{\mathbf{Z}}$ se verifica que

$$p(\mathbf{x}|\mathbf{z}, \mathbf{y}) = p(\mathbf{x}|\mathbf{z})$$

De lo contrario se dice que $\mathbf{X}$ e $\mathbf{Y}$ son condicionalmente dependientes dado $\mathbf{Z}$. Cuando $\mathbf{X}$ e $\mathbf{Y}$ son condicionalmente independientes dado $\mathbf{Z}$ se nota como $I(\mathbf{X}, \mathbf{Y}|\mathbf{Z})_p$

***Teorema de Bayes***: este teorema nos permite representar la probabilidad condicionada $p(y|x)$ mediante la siguiente expresión:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

Teniendo en cuenta que $p(x) = \sum_{y \in \Omega_Y} p(x, y)$ y que $p(x, y) = p(x|y)p(y)$, podemos representar el teorema de Bayes usando la siguiente expresión:

$$p(y|x) = \frac{p(x|y)p(y)}{\sum_{y \in \Omega_Y} p(x|y)p(y)} \tag{1.1}$$

En la ecuación 1.1 podemos distinguir:

- La probabilidad $p(y)$ se denomina probabilidad *marginal*, *a priori*, o *inicial* de $Y = y$ puesto que puede ser obtenida antes de conocer la evidencia, es decir, no tiene en cuenta ninguna información acerca de $X = x$.

- La probabilidad $p(y|x)$ es la probabilidad *posterior*, *a posteriori*, o *condicional* de $Y$ puesto que se obtiene después de conocer la evidencia, es decir, depende del valor $x$.

- La probabilidad $p(x|y)$ se le llama *verosimilitud* y es la probabilidad de la observación $X = x$ dado $Y = y$.

## 1.3. Redes bayesianas.

Considérese un conjunto finito $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$ de variables aleatorias discretas, cada variable $X_i$ toma valores dentro de un conjunto finito $\Omega_{X_i}$. Una *red bayesiana* es una representación gráfica de la distribución de probabilidad conjunta; más formalmente, es un par $(G, \Theta)$, donde $G$ es un grafo dirigido acíclico (en inglés, *directed acyclic graph*, también llamado *DAG*) cuyos nodos se corresponden con las variables aleatorias de $\mathbf{X}$ y $\Theta$ son los parámetros que especifican las distribuciones de probabilidad [263]. Por tanto, podemos decir, que una red bayesiana está compuesta por una componente cualitativa y otra cuantitativa.

**Componente cualitativa.** Compuesta por un grafo dirigido acíclico $G = (\mathbf{X}, E_G)$, donde $\mathbf{X}$, es el conjunto de nodos, representando las variables del sistema y $E_G$, el conjunto de arcos, representando relaciones de dependencia directa entre variables. Cuando tenemos dos variables $X_1$ y $X_2$ conectadas por un arco $X_1 \rightarrow X_2$ podemos deducir que ambas variables están relacionadas. Cuando dicho arco no existe podemos decir que existe una relación de independencia (marginal o condicional) entre $X_1$ y $X_2$.

En la figura 1.1 se ilustra el grafo dirigido acíclico de una red bayesiana para la gripe A (H1N1) de 2009, donde tenemos variables como *Tos* que puede tomar los valores *Ausente*, *Leve* o *Pronunciada*, o la variable *Gripe A* que puede ser *Presente* o *Ausente*, o la variable *Fiebre* que se mueve en el intervalo de los 36 grados a los 42, en pasos de 0,5 grados.

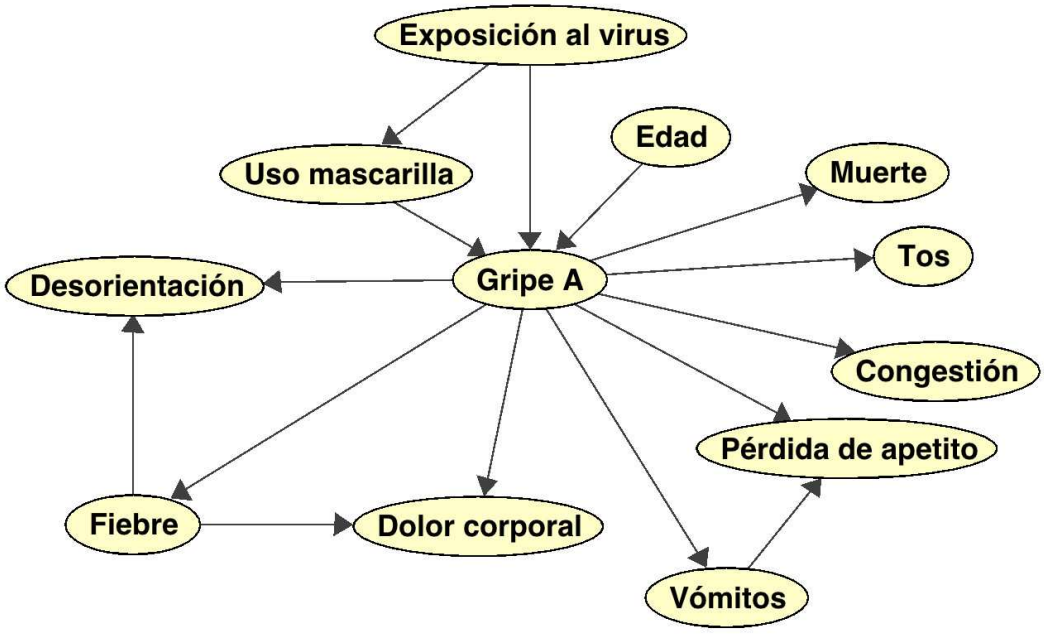


Figura 1.1: Ejemplo de red bayesiana para la gripe A.

Para dotar de una interpretación semántica completa a un grado dirigido acíclico se necesita un criterio que determine, de forma precisa, que propiedades de independencia son reflejadas por la topología de red, para ello utilizaremos el criterio de *d-separación* o *separación dirigida* [263], pero haremos una definición previa.

***Definiciones***: un *camino* no dirigido es una secuencia de nodos conectados por enlaces en el grafo (el camino puede pasar por un enlace en cualquiera de sus sentidos). Un *enlace cabeza-cabeza* en un nodo de un camino se produce cuando el camino que pasa por dicho nodo tiene la forma $X \rightarrow Z \leftarrow Y$. El nodo $Z$ es un nodo cabeza-cabeza en el camino. Un camino se dice *activo* por un conjunto de variables $\mathbf{Z}$ si se satisface que:

- Todo nodo del camino con arcos cabeza-cabeza está en $\mathbf{Z}$ o tiene algún descendiente dentro de $\mathbf{Z}$.
- Cualquier otro nodo en el camino no pertenece a $\mathbf{Z}$.

Si no se satisface esta relación se dice que el camino está *bloqueado* por $\mathbf{Z}$.

***Criterio d-separación***: sean $\mathbf{X}$,$\mathbf{Y}$ y $\mathbf{Z}$ tres subconjuntos disjuntos de variables en un grafo dirigido acíclico $G$, entonces se dice que $\mathbf{Z}$ d-separa $\mathbf{X}$ de $\mathbf{Y}$, o lo que es lo mismo $\mathbf{X}$ e $\mathbf{Y}$ son independientes dado $\mathbf{Z}$ y lo notamos como $I(\mathbf{X}, \mathbf{Y}|\mathbf{Z})_G$, si y sólo si todos los caminos entre cualquier nodo de $\mathbf{X}$ y cualquier nodo de $\mathbf{Y}$ están bloqueados por $\mathbf{Z}$.

Otro patrón gráfico de independencia interesante (y de gran utilidad como se verá posteriormente en el apartado 1.5) es el generado por el *manto de Markov* [263]. El manto de Markov de una variable $X$ en $G$, $MB_G(X)$, es el conjunto de variables compuesto por los padres de $X$, los hijos de $X$ y los padres de los hijos de $X$ en $G$. Este conjunto tiene la propiedad de que $X$ es independiente del resto de las variables (aquellas en $\mathbf{X}\backslash(MB(X)\cup\{X\})$) dado su manto de Markov, más formalmente, $I(X, \mathbf{X} \setminus (MB(X) \cup \{X\})|MB(X))$.

En el ejemplo de la figura 1.1 podemos ver que la variable *Gripe A* está relacionada con *Exposición al virus*. Podemos observar que *Fiebre*, *Desorientación*, *Dolor Corporal*, *Vómitos*, *Pérdida de apetito*, *Congestión* y *Tos* son condicionalmente independientes de *Exposición al virus* y *Uso mascarilla*, dada la variable *Gripe A*. También se puede ver que la variable *Edad* es marginalmente independiente a *Exposición al virus*. Asimismo podemos obtener que el manto de Markov para la variable *Gripe A* está compuesto por todas las variables, mientras que, por ejemplo, el manto de Markov de la variable *Dolor corporal*, solamente está compuesto por las variables *Gripe A* y *Fiebre*.

Las relaciones entre los nodos de un arco en una red bayesiana se puede ver como una relación causa-efecto. La variable a la que apunta el arco es dependiente de la que está en el origen de éste. En nuestra figura 1.1 se observa claramente que los distintos síntomas que aparecen son causados por la gripe.

**Componente cuantitativa.** Compuesta por una colección de parámetros numéricos para cada variable en $\mathbf{X}$, normalmente tablas de probabilidad condicionada que expresan la creencia que tenemos sobre las relaciones entre las variables. Para cada variable $X \in \mathbf{X}$ tenemos una familia de distribuciones condicionadas $p(X|pa_G(X))$, una para cada *configuración*, $pa_G(X)$, del *conjunto de padres* de $X$ en el grafo, $Pa_G(X) = \{Y \in \mathbf{X} \mid Y \rightarrow X \in E_G\}$. A partir de estas distribuciones condicionales podemos recuperar la distribución conjunta sobre $\mathbf{X}$:

$$p(x_1, x_2, \ldots, x_n) = \prod_{X_i \in \mathbf{X}} p(x_i|pa_G(X_i)) \tag{1.2}$$

En la figura 1.2 podemos ver la tabla de probabilidad condicionada de la variable *Muerte* dada su variable padre *Gripe A*. Como se puede observar la mortandad de esta enfermedad es del 1,2 % [2].


Figura 1.2: Tabla de probabilidad condicionada para la variable *Muerte* en la red bayesiana para la gripe A.

La expresión (1.2) representa la descomposición de la distribución conjunta en productos de factores más simples. Las relaciones de dependencia/independencia que hacen posible esta descomposición, están representadas en el grafo (a través del criterio de d-separación) mediante la presencia o ausencia de conexiones directas entre pares de variables en el grafo $G$.

Por todo lo visto, las redes bayesianas, como modelo, son visualmente interpretables ya que su componente gráfica refleja de forma explícita las relaciones entre las variables de un dominio. Obsérvese que este tipo de relaciones causa-efecto, se aproximan a la forma que tiene el ser humano de estructurar el conocimiento. Además, la naturaleza probabilística de las redes

[2]Según los datos de la Organización Mundial de la Salud en su informe del día 11 de Octubre de 2009 (disponible en `http://www.who.int/csr/don/2009_10_16/en/index.html`). No obstante, en España, la tasa de letalidad es de 0,16 por cada mil afectados; según el informe del día 22 de Octubre de 2009 del Ministerio de Sanidad (disponible en `http://www.msc.es/servCiudadanos/alertas/informesGripeA/091022.htm`).

bayesianas les permite representar nuestra incertidumbre sobre un problema. Podemos decir que, entre otras, facilitan varias tareas, como:

- La *adquisición* del conocimiento: proceso por el cual se detecta y se representa el conocimiento del experto, ya que se puede codificar información acerca de la dependencia e independencia de variables de forma gráfica antes de tener que entrar a cuantificar las probabilidades. Además, debido a la modularización del modelo, se reduce considerablemente el número de parámetros a considerar. Por tanto, las redes se pueden construir manualmente con la ayuda de un experto (como veremos en la sección 1.4.1) y además se pueden construir de forma automática a partir de datos (apartado 1.4.2).

- La *explicación e interpretación*: al ser gráficamente interpretable, se simplifica el proceso de comunicación con el experto o la obtención de información relevante. Y además, una vez obtenida la red, se puede realizar el proceso inverso, es decir, se puede extraer conocimiento de la misma (ver sección 1.3.1.2).

- La *predicción*: nos permiten la generación de predicciones basada en información aportada a posteriori, por ejemplo, en la red de la figura 1.1, conociendo los síntomas, podemos ver la probabilidad que un sujeto presente gripe o no. Esto nos será útil en problemas de clasificación supervisada (apartado 1.5). Siguiendo el ejemplo, podemos conocer si un nuevo paciente padece la enfermedad a partir de los síntomas. Igualmente interesante, es que permite presentar al usuario un conjunto de posibles alternativas, ordenadas desde la más probable a la menos probable.

- La *actualización*: la llegada de nuevas observaciones puede incorporarse de forma fácil a la red, permitiéndonos obtener nuevas conclusiones conforme nos llegan nuevos datos, como veremos en la sección 1.3.1.1.

- El *estudio de series temporales*: con el formalismo de las redes bayesianas dinámicas (apartado 1.3.2) podemos representar relaciones temporales entre las variables.

### 1.3.1. Probabilidad condicional *a posteriori*.

#### 1.3.1.1. Propagación.

Cuando disponemos de una red bayesiana necesitamos obtener nuevas conclusiones a medida que vamos obteniendo nueva información, o *evidencia*. Por ejemplo, si nuestra red bayesiana nos permite diagnosticar algún tipo de enfermedad a partir de los síntomas del paciente (evidencia), si se presentan nuevas variantes de la enfermedad (nuevas evidencias) necesitaremos actualizar el conocimiento que nuestra red modeliza. El mecanismo para obtener conclusiones a partir de la evidencia se conoce como *propagación de la evidencia* o, de manera más simple, *propagación* [263, 310, 311], aunque algunos autores utilizan otra terminología como *propagación de la incertidumbre* o *inferencia probabilística*.

Podemos entonces ver que la propagación consiste en realizar los cálculos necesarios para obtener la probabilidad a posteriori de una o varias variables dados los valores asignados a otras variables en la red bayesiana (evidencia $\mathbf{E}$), es decir, el cálculo de $P(\mathbf{X}_i|\mathbf{E})$, donde $\mathbf{X}_i$ y $\mathbf{E}$ son una única variable o un conjunto de ellas. Por ejemplo, en la figura 1.3 podemos observar la probabilidad a posteriori de la variable *Muerte* cuando la evidencia de *GripeA* es igual a *Presente*.

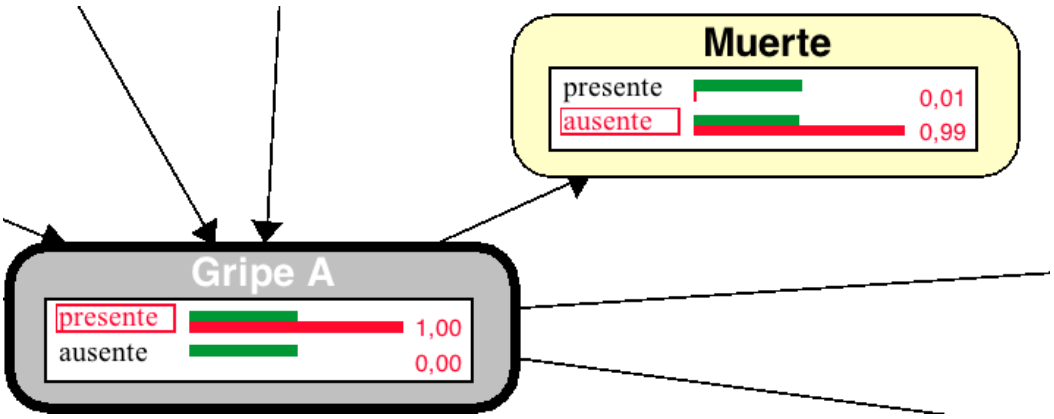


Figura 1.3: Probabilidad de la variable *Muerte*, teniendo la evidencia de *GripeA* igual a *Presente*.

Cooper [112] demostró que el calculo de la probabilidad a posteriori es un problema NP-difícil (es #P-completo), incluso para una única variable y, por tanto, cualquier tipo de inferencia probabilística es también un problema con una complejidad exponencial. Por este motivo, hay varias propuestas para realizar la propagación de la evidencia en las redes bayesianas:

- *Métodos exactos* [153]: es una aproximación que calcula las probabilidades a posteriori sin otro error que el redondeo producido en los cálculos. Al ser un problema de complejidad exponencial sólo se utilizan cuando calculan la propagación en un intervalo de tiempo asumible, lo cual se produce sólo para algunos tipos especiales de redes bayesianas.

- *Métodos aproximados* [297]: cuando no es posible la propagación exacta de probabilidades en un tiempo razonable, se utilizan métodos aproximados que pierden exactitud en la solución a cambio de obtener los resultados en un tiempo mucho menor. Estos métodos utilizan distintas técnicas de simulación con valores aleatorios para obtener valores aproximados de las probabilidades, o bien, utilizan técnicas deterministas y aproximadas que siempre llegan a la misma solución.

#### 1.3.1.2. Abducción.

El problema de la *abducción* [124] en las redes bayesianas puede verse como la búsqueda de explicaciones a partir de una red dada. El enfoque clásico de la abducción en lógica es el siguiente: Si la sentencia $A \rightarrow B$ es verdadera y B es verdadera, entonces A es posiblemente verdadera. Por tanto en abducción, se empieza por una conclusión y se procede a derivar las condiciones que podrían hacer a esta conclusión válida. En otras palabras, el resultado de la abducción es una *hipótesis* o posible explicación y no una conclusión.

Como en el proceso de abducción se suele producir siempre más de una explicación posible, es necesario quedarse con aquellas que sean más plausibles. Los criterios que se suelen utilizar para seleccionar las mejores explicaciones se basan en alguna medida que nos diga cuando una hipótesis es mejor que otra, por ejemplo, se tiende a escoger explicaciones más sencillas.

La abducción en redes bayesianas, también denominada *revisión de creencias* o *búsqueda de la explicación más probable* por Pearl [262] y *búsqueda de*

*la configuración máxima a posteriori* por Shimony y Charniak [312] consiste en encontrar la configuración de estados de mayor probabilidad para las variables no observadas. En general, una explicación no es suficiente y se buscan las $K$ explicaciones más probables de los hechos observados.

Se distinguen dos tipos de abducción en las redes bayesianas, la abducción total (sobre todas las variables) y la abducción parcial (sobre un subconjunto de variables). La abducción parcial que puede verse como una generalización de la total es más interesante en su utilización práctica pero presenta más problemas para ser resuelta de manera eficiente [124].

### 1.3.2. Redes bayesianas dinámicas.

El formalismo de las redes bayesianas dinámicas [133, 239] nos permite aplicar redes bayesianas a dominios dinámicos, es decir, nos permiten modelar datos temporales, creando una réplica de cada variable para cada instante. Una de las ventajas que presentan es la capacidad de representar ciclos, al contrario, de las redes bayesianas estáticas. Las redes bayesianas dinámicas son redes sin ciclos, no obstante, se pueden representar ciclos en las relaciones causa-efecto entre las variables con distintos instantes de tiempo. Las variables del instante $t$ influyen en las variables del instante $t + 1$, como se puede apreciar en la figura 1.4.

En las redes bayesianas dinámicas primero se considera una red estática a partir de la cual se genera una copia de cada modelo estático para cada uno de los intervalos de tiempo estudiados. Finalmente se establecen enlaces entre nodos pertenecientes a redes bayesianas estáticas consecutivas. A estos enlaces se les denomina *arcos temporales* o *relaciones temporales* de un periodo de tiempo $t$ y definen como la distribución de variables del periodo $t$ son dadas condicionalmente sobre la distribución de variables del periodo de tiempo $t - 1$. Se asume que las redes bayesianas dinámicas cumplen la propiedad de Markov, es decir, el futuro es independiente del pasado dado el presente.

En la figura 1.4 se puede ver un ejemplo de red bayesiana dinámica, donde distinguimos dos redes bayesianas estáticas para dos intervalos. Las unen tres arcos temporales, notados en rojo en la figura.

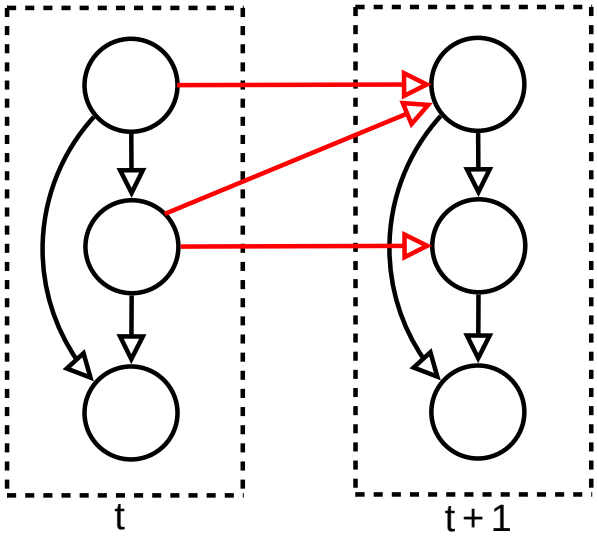


Figura 1.4: Ejemplo de red bayesiana dinámica, donde los arcos temporales están en rojo.

Resumiendo, una red bayesiana dinámica se puede considerar un grafo dirigido acíclico, donde el conjunto de variables se repiten en el tiempo y el conjunto de arcos está compuesto por arcos temporales que interconectan periodos de tiempo consecutivos y arcos no temporales que interconectan las variables en el mismo periodo de tiempo.

## 1.4. Obtención de redes bayesianas.

Para utilizar una red bayesiana es necesario construirla antes. La obtención de la misma se puede hacer, principalmente, mediante dos métodos: uno, se puede construir de forma manual, elicitándola del experto, esto es, el experto indica las relaciones que hay entre las variables y las tablas de probabilidad asociada, o construirla de forma automática, es decir, se obtienen las relaciones y sus tablas de probabilidad condicionada de forma automática a partir de un conjunto de datos.

Para el primer caso necesitamos el conocimiento de las distintas variables del problema y de sus relaciones, por lo que necesitamos un experto en la

materia que represente ese conocimiento en una red bayesiana.

Para el segundo caso, necesitamos distintas técnicas de aprendizaje automático de redes bayesianas, que analicen los datos[3], extraigan las relaciones entre las distintas variables que componen el dominio del problema y las representen mediante una red.

Así mismo se puede considerar una tercera posibilidad mixta entre las dos anteriores y es que, utilicemos algún tipo de información de un experto a la hora de realizar el aprendizaje automático de la red.

### 1.4.1. Redes bayesianas elicitadas por el experto.

Una persona puede analizar un problema y extraer el conocimiento de forma artesanal, plasmándolo en un programa o, mejor aún, si disponemos de alguien familiarizado con la materia -un experto-, podemos aprovechar su conocimiento directamente, sin tener que analizar los datos. Esta forma de obtención de redes bayesianas tiene serios inconvenientes. El análisis humano de los datos no siempre es factible debido a que la cantidad de información a procesar sea demasiado grande, lo cual, es lo más común en la mayoría de los casos. Si trabajamos con un experto en la materia en lugar de analizar los datos directamente, nos encontramos que no es fácil extraer el conocimiento del experto y que incluso, el conocimiento obtenido no sea del todo objetivo.

La construcción manual de redes bayesianas [96, 223] normalmente implica varias etapas diferenciadas que podemos simplificar en dos fases. Una primera, de recopilación de información cualitativa, esto es, identificar las variables relevantes dentro del dominio de estudio y las relaciones de independencia/dependencia existentes entre ellas, formando una red con dicha información. Y una segunda etapa donde se busca obtener una información cuantitativa de la red bayesiana, esto es, obtener las probabilidades a priori y las probabilidades condicionales. Obsérvese que cada etapa se corresponde con cada una de las partes en las que podemos diferenciar una red bayesianas, la parte cualitativa -un grafo dirigido acíclico- y la parte cuantitativa -las tablas de probabilidad asociadas. Entremos un poco más en detalle y veamos los pasos necesarios para construir una red bayesiana de forma manual:

[3]Los datos se suelen almacenar de forma tabular: en las columnas tenemos las distintas variables y en las filas los distintos casos, esto es, cada fila representa un experimento o muestra y, dentro de esa fila, cada valor equivale a su correspondiente variable: el primer valor para la primera variable, el segundo valor para la segunda variable, etc.

- *Parte Cualitativa*: para la construcción o edición del grafo de la red bayesiana, el primer paso que tenemos que dar es la identificación de aquellas variables más importantes junto con el conjunto de valores que pueden tomar cada una de ellas. Dicha obtención de variables es realizada por un ingeniero en conocimiento y se hace generalmente preguntándole al experto y estudiando el dominio del problema a resolver. Una vez decididas las variables a incluir en el grafo hay que incorporar las relaciones de dependencia e independencia existentes entre ellas. Para este segundo cometido normalmente se hace uso de la noción de causalidad, si hay una relación de causa-efecto entre dos variables se establece un arco entre dichas variables.

- *Parte Cuantitativa*: esta parte está compuesta por las tablas de probabilidad condicionada entre las variables y las probabilidades a priori de las mismas. La obtención de los datos numéricos se hace a partir del conocimiento de los expertos. Este conocimiento suele ser subjetivo y con un número de valores a estimar bastante alto. Respecto a la subjetividad, supongamos un suceso poco probable, el experto nos puede dar una estimación de $0,001$ ó $0,01$. Como ejemplo de complejidad, en el sistema experto *Pathfinder III* para el diagnóstico de enfermedades linfáticas [147], donde el número de síntomas del dominio es 112 y el número de casos para la variable Enfermedad es 60, fue necesario establecer un total de 16.620 parámetros por parte de un experto, lo que supuso aproximadamente 40 horas de elicitación de valores para los parámetros. En estos casos, como alternativa se puede aplicar algún método de aprendizaje automático de las tablas de probabilidad si se dispone de los datos necesarios.

La obtención manual de una red bayesiana no deberá verse como un proceso secuencial, sino más bien como un proceso iterativo donde, en cada paso, se va refinando el modelo y adaptándolo a nuevos casos. También se debe considerar que es un proceso complejo y largo con bastante subjetividad. Además hay que tratar con otra serie de problemas, como es la poca disponibilidad de expertos sobre algún tema o la necesidad de encontrar distintos expertos especializados en distintos dominios del problema. También es un obstáculo la falta de formación de los expertos en probabilidad y las dificultades con el manejo de las relaciones de independencia condicional.

### 1.4.2. Aprendizaje automático de redes bayesianas.

Como se ha comentado, la construcción de una red elicitada por un experto es una tarea difícil y lenta, y el modelo obtenido sesgado. Por ello se planteó el aprendizaje de la red a partir de los datos, el aprendizaje automático. Al igual que en la construcción manual de una red bayesiana, dentro del aprendizaje automático encontramos dos tareas diferenciadas aunque bastante relacionadas entre sí. Dado que una red bayesiana se compone de una componente cualitativa y otra cuantitativa, distinguimos dos tipos de aprendizaje automático:

- *Aprendizaje estructural*: el aprendizaje de la estructura gráfica (un grafo dirigido acíclico), es decir, la parte cualitativa.
- *Aprendizaje paramétrico*: el aprendizaje de los parámetros numéricos (las tablas de probabilidad de cada nodo del grafo), es decir, la parte cuantitativa.

Estos dos aprendizajes no se pueden realizar de forma totalmente independiente, dado que para poder estimar las distribuciones de probabilidad asociadas a cada nodo de la red, nos hará falta conocer la topología de la red, que es lo que nos va a modularizar los parámetros por familias, por cada nodo la distribución de probabilidad dados sus padres, determinando así el número de parámetros necesarios. Por ejemplo en la red de la gripe A (figura 1.1) si contiene el arco $GripeA \rightarrow Congestión$, sabemos que tenemos que calcular los valores $P(Congestión = presente|Gripe = presente)$ de la distribución de probabilidad de $Congestión$ dado $GripeA$.

Además, para poder determinar si hay o no un enlace entre dos nodos dados, tendremos que cuantificar numéricamente la relación entre esas variables.

En los siguientes subapartados veremos los distintos métodos que hay para el aprendizaje de redes bayesianas, tanto de la estructura gráfica como del los parámetros numéricos.

#### 1.4.2.1. Aprendizaje paramétrico.

Una vez que tenemos la estructura de la red -en el siguiente apartado veremos cómo- tendremos que determinar los parámetros asociados a cada nodo del grafo a partir de los datos de los que disponemos. Por tanto, partimos de un grafo acíclico dirigido $G = (\mathbf{X}, E_G)$, donde $\mathbf{X}$, es el conjunto de

nodos, representando las variables del sistema y $E_G$, el conjunto de arcos. La estructura de red determinará para cada variable $X_i \in \mathbf{X}$, su conjunto de padres, que denotamos como $Pa_G(X_i)$ , $Pa_G(X_i) = \{Y \in \mathbf{X} \mid Y \rightarrow X_i \in E_G\}$, por lo que la distribución de probabilidad conjunta puede obtenerse de la siguiente manera:

$$p(x_1, x_2, \ldots, x_n) = \prod_{X_i \in \mathbf{X}} p(x_i | pa_G(x_i))$$

donde $x_i$ representa un valor de la variable $X_i$ y $pa_G(x_i)$ representa una asignación de valores a cada una de las variables del conjunto $Pa_G(X_i)$. Entonces el problema consiste en estimar los valores de las distribuciones de probabilidad condicionadas $P(x_i|pa_G(x_i))$ a partir de los datos disponibles.

La forma habitual y más simple de estimar las distribuciones de probabilidad condicionadas es mediante el cálculo de las frecuencias relativas de ocurrencia de los correspondientes sucesos. Por tanto, si $n(pa_G(x_i))$ y $n(x_i, pa_G(x_i))$ representan respectivamente el número de casos de la base de datos en que las variables de $Pa_G(X_i)$ toman los valores $pa_G(x_i)$ y en que las variables $X_i$ y $Pa_G(X_i)$ toman simultáneamente los valores $x_i$ y $pa_G(x_i)$, entonces el valor estimado de la probabilidad es:

$$p(x_i | pa_G(x_i)) = \frac{n(x_i, pa_G(x_i))}{n(pa_G(x_i))}$$

En términos más formales, este método de estimar la probabilidad corresponde con la utilización de un estimador de máxima verosimilitud [58]. Dicha estimación tiene dos inconvenientes: datos dispersos y sobreajuste. Cuando se tienen datos dispersos, el estimador $n(pa_G(x_i))$ es cero. El segundo inconveniente es que el estimador por máxima verosimilitud tiende a sobreajustarse a los datos; cuando el tamaño muestral es lo suficientemente grande este estimador tiende al valor verdadero, pero para muestras pequeñas las estimaciones se sobreajustan a los datos disponibles y se tiene poca capacidad de generalización.

Se han planteado distintos métodos de estimación de probabilidades que intentan solventar estos problemas. El primero que vamos a ver está basado en la *Ley de la sucesión de Laplace* [137], que nos dice que si en una muestra de $N$ casos encontramos $k$ casos que verifican una determinada propiedad $Q$,

por ejemplo, que la variable $X$ es igual a $x_i$, entonces la probabilidad de que el siguiente caso que observamos tenga la misma propiedad es $(k+1)/(N+|Q|)$, donde $|Q|$ representa el número de alternativas posibles que se consideran para la propiedad Q, en el ejemplo anterior, el número de distintos valores posibles que puede tomar $X$. En nuestro caso para estimar la probabilidad $p(x_i|pa_G(x_i)))$ con este método usaríamos:

$$p(x_i|pa_G(x_i)) = \frac{n(x_i, pa_G(x_i)) + 1}{n(pa_G(x_i)) + |X_i|}$$

donde $|X_i|$ es el número de distintos valores que la variable $X_i$ puede tomar. Nótese que para muestras muy grandes, las diferencias de éste método con respecto el estimador de máxima verosimilitud serán pequeñas, en cambio, cuando las muestras son muy pequeñas la distribución tiende hacia una distribución uniforme. Obsérvese que en el caso que $n(pa_G(x_i))$ sea cero, el resultado es exactamente la distribución uniforme. Este método de estimación es un método realmente bayesiano, ya que se parte de cierta información a priori sobre el parámetro -la distribución uniforme si la variable es binaria- y se actualiza dicha información con los nuevos datos empleando la fórmula de Bayes.

Otro método con el cual podemos paliar los problemas de sobreajuste y datos dispersos es usar un método de estimación bayesiano más general, basado en distribuciones de Dirichlet [150]. Sin entrar en demasiada profundidad, vamos a mostrar este estimador bayesiano, donde suponemos que las distribuciones a priori son Dirichlet, para estimar la probabilidad $p(x_i|pa_G(x_i))$. Con este método usaríamos:

$$p(x_i|pa_G(x_i)) = \frac{n(x_i, pa_G(x_i)) + \frac{s}{|X_i|}}{n(pa_G(x_i)) + s}$$

donde $s$ es un parámetro que se suele interpretar en términos de *tamaño muestral equivalente*, es decir, es como si la distribución a priori se hubiese estimado a partir de una muestra de tamaño $s$. Una formulación equivalente, pero quizás más intuitiva, es la siguiente [118]:

$$p(x_i|pa_G(x_i)) = \alpha \frac{n(x_i, pa_G(x_i))}{n(pa_G(x_i))} + (1 - \alpha)\frac{1}{|X_i|}$$

donde $\alpha = \frac{n(pa_G(x_i))}{n(pa_G(x_i))+s}$. En otras palabras, el estimador es el resultado de realizar una combinación convexa entre la probabilidad condicional de $X_i$ dados sus padres y la probabilidad uniforme estimada por máxima verosimilitud.

#### 1.4.2.2. Aprendizaje estructural.

En esta etapa del proceso de aprendizaje automático tenemos que buscar las relaciones cualitativas entre las variables del problema. Tengamos en cuenta que el conjunto de redes bayesianas con $n$ nodos es de orden super-exponencial[4][290], con lo que un recorrido exhaustivo por dicho conjunto con el fin de encontrar la mejor red candidata no es factible en la mayoría de los casos. Además se ha demostrado que el problema del aprendizaje es un problema NP-difícil [75].

Podemos realizar la siguiente clasificación de las estrategias de aprendizaje en base a la técnica utilizada para obtener la parte cualitativa de la red.

- *Basadas en tests de independencia*: son métodos que utilizan criterios de independencia entre variables, para obtener la estructura que mejor representa el conjunto de independencias que se deducen de los datos.

- *Métricas + búsqueda*: son paradigmas de aprendizaje que se basan en un criterio de la bondad del ajuste de una estructura a los datos. Utilizando dicho criterio se realiza un proceso de búsqueda entre las estructuras candidatas, dando como resultado aquella estructura que mejor se ajuste a los datos.

- *Híbridos*: son modelos que combinan ideas de las anteriores metodologías.

La idea subyacente en el segundo tipo de métodos, es encontrar el grafo que mejor represente los datos, utilizando el menor número de arcos posibles, es decir, la calidad de cada grafo candidato se cuantifica mediante algún tipo de medida o métrica. Dicha medida es utilizada por un algoritmo de búsqueda para encontrar las mejores soluciones desde el punto de vista de la medida utilizada. Por tanto, estos métodos se caracterizan tanto por la métrica usada como por el algoritmo de búsqueda.

[4]El número de grafos dirigidos acíclicos posibles para n nodos sería: $f(n) = \sum_{i=1}^{n}(-1)^{i+1}\frac{n!}{i!(n-i)!}2^{i(n-i)}f(n-i)$. Por ejemplo, $f(3) = 25$, $f(5) = 25000$, $f(10) \approx 4{,}2 \cdot 10^{18}$.

La intención de los métodos basados en tests de independencia es llevar a cabo un estudio cualitativo de las relaciones de independencia condicionales que existen entre las variables usando para ello los datos disponibles. La idea es buscar el grafo que mejor represente las relaciones de independencia condicional entre conjuntos de variables.

También podemos encontrar lo que denominaremos aproximaciones híbridas que son métodos que utilizan una combinación de ambas técnicas, es decir, utilizan un algoritmo de búsqueda guiado por una métrica y la detección de independencias entre las variables.

**Aprendizaje automático basado en tests de independencia.**

Los métodos de aprendizaje basados en tests de independencia [68, 71, 84, 89, 130, 230, 321, 345, 346, 354] son en cierta forma independientes del modelo de representación cuantitativa expresada en una red, debido a que, como hemos visto, su objetivo no es obtener una red donde la distribución de probabilidad que representa se parezca a la original de la que provienen los datos, sino que hacen un estudio cualitativo de las relaciones de independencia que podemos encontrar en los datos y a partir de él, intentan recuperar la red que mejor represente dichas relaciones.

La información de entrada que necesitan este tipo de algoritmos, son las relaciones de independencia condicional entre conjuntos de variables, y la información de salida que devuelven es un grafo que representa la mayor parte o todas las relaciones. Lo habitual es que tengamos como entrada una base de datos en lugar de una lista de relaciones de independencia condicional, por tanto, lo que se hace es una serie de tests estadísticos de independencia condicional entre variables.

Un problema que tienen estos algoritmos de aprendizaje, es el coste computacional asociado a los tests que realizan, ya que aumenta conforme aumenta el tamaño del conjunto condicionante: exponencial al tamaño. Otro inconveniente que presentan es que se necesita una gran cantidad de datos para que la detección de independencias sea fiable.

Los métodos de aprendizaje basados en test de independencias normalmente empiezan con un grafo completo no dirigido, es decir, un grafo donde to-

dos los nodos están relacionados con todos y proceden a eliminar enlaces que unen pares de nodos que son condicionalmente independientes dado un subconjunto de nodos. Posteriormente orientan dichos arcos formando patrones cabeza-cabeza (que son tripletas de nodos $X$,$Y$,$Z$ donde $X$ e $Y$ son no adyacentes y existen los arcos $X \rightarrow Z$ e $Y \rightarrow Z$). Por ejemplo, los algoritmos PC [320] y SGS [321] intentan primero eliminar tantos enlaces como pueden, y después orientan los enlaces que quedan del grafo formando patrones cabeza-cabeza. Finalmente orientan los enlaces que queden siguiendo algunas normas de coherencia.

**Aprendizaje automático: métricas+búsqueda.**

Este tipo de estrategia (en inglés, *score+search*) trata de encontrar la estructura que maximiza una métrica dada. Dicha métrica se define como una medida de cómo una estructura se ajusta a los datos. Por tanto, los métodos basados en este paradigma requieren de una función de ajuste o métrica y una forma de explorar el espacio de búsqueda para evaluar la aptitud de una estructura candidata, mientras que el algoritmo de búsqueda se mueve entre distintas posibles soluciones. El objetivo es devolver aquella estructura que se ajuste mejor a los datos, de cuantas se hayan evaluado. Este tipo de métodos se puede ver como un problema clásico de optimización donde tenemos que tener en cuenta:

- *Métrica*: se basa en alguna que nos permite determinar el grado de ajuste de la estructura y los datos. Los tipos de métrica que se han utilizado se pueden englobar dentro de las siguientes categorías: medidas de información y métricas bayesianas.

- *Método de búsqueda*: una vez que tenemos una medida para evaluar cada estructura, necesitamos un método de búsqueda que nos permita explorar el espacio de búsqueda entre las configuraciones vecinas, con el objetivo de encontrar la mejor estructura. La mayoría de los algoritmos de búsqueda son de tipo voraz (*greedy*), debido al tamaño super-exponencial del espacio de búsqueda.

**Métricas.**

El problema de aprender una red usando un enfoque métrica+búsqueda se puede definir como: dado un conjunto de datos $D$ completo[5], compuesto por $D = \mathbf{x}_1, .., \mathbf{x}_N$ instancias de $\mathbf{X}$, tenemos que encontrar un grafo dirigido acíclico $G^*$, que verifique:

$$G^* = arg\, max_{G \in \mathcal{G}_n} g(G : D)$$

donde $g(G : D)$ es la métrica que mide el grado de ajuste de cualquier grafo candidato $G$ al conjunto de datos $D$, y $\mathcal{G}_n$ es la familia de todos los grafos dirigido acíclicos definidos para $\mathbf{X}$.

Hay distintas métricas, la mayoría pueden ser agrupadas en dos categorías: bayesianas y basadas en medidas de información.

Para hacer un repaso de las mismas se va a seguir la siguiente notación: el número de estados de la variable $X_i$ es $r_i$, el número de posibles configuraciones del conjuntos de padres $Pa_G(X_i)$ de $X_i$ es $q_i$; $N_{ijk}$ es el número de instancias del conjunto de datos $D$ donde la variable $X_i$ toma el valor $x_{ik}$ y el conjunto de variables $Pa_G(X_i)$ toma el valor $\mathbf{w}_{ij}$; $N_{ij}$ es el número de instancias en $D$ que las variables de $Pa_G(X_i)$ toman en su j-ésima configuración $\mathbf{w}_{ij}$; obviamente $N_{ij} = \sum_{k=1}^{q_i} N_{ijk}$; análogamente, $N_{ik}$ es el número de instancias de $D$ donde la variable $X_i$ toma su k-ésimo valor $x_{ik}$ y, por lo tanto, $N_{ik} = \sum_{j=1}^{q_i} N_{ijk}$; el número total de instancias en $D$ es $N$.

Las **métricas bayesianas** [57, 58, 80, 121, 117, 127, 149, 150, 225, 282, 325] buscan la estructura que maximiza la probabilidad de una red condicionada a la base de datos $p(G|D)$, usando para ello la fórmula de Bayes:

$$p(G|D) = \frac{p(D|G)p(G)}{p(D)}$$

Como los datos son siempre los mismos para las distintas redes de un mismo problema, el denominador $p(D)$ de la anterior expresión puede ignorarse. El término $p(G)$ representa la distribución *a priori* de cada estructura candidata; en muchos casos se utiliza una distribución uniforme por lo que también puede ignorarse. En la práctica, al ser más cómodo trabajar en el espacio logarítmico, las métricas calculan $log\, p(G|D)$ en lugar de $p(G|D)$. Finalmente,

[5]En el sentido de que carece de valores perdidos o desconocidos.

el término $p(D|G)$, llamado *evidencia*, es la verosimilitud muestral promedio que puede calcularse bajo ciertas suposiciones (diferentes suposiciones dan lugar a diferentes métricas).

Por ejemplo, en la métrica K2 [80] si se dan una serie de condiciones (independencia de los casos de la base de datos, inexistencia de casos con datos perdidos o desconocidos, uniformidad de las distribuciones de probabilidad de los parámetros de una red dada ésta) se puede deducir una fórmula que establece cuál es la distribución de probabilidad conjunta de una estructura $G$ y una base de datos $D$:

$$g_{K2}(G:D) = log(p(G)) + \sum_{i=1}^{n} \left[ \sum_{j=1}^{q_i} \left[ log\left( \frac{(r_i-1)!}{(N_{ij}+r_i-1)!} \right) + \sum_{k=1}^{r_i} log(N_{ijk}!) \right] \right]$$

Debido a que la métrica K2 impone la uniformidad de la distribución *a priori* sobre los parámetros, Heckerman y col. [150] propusieron la métrica BD (del inglés, *Bayesian Dirichlet*), que se puede considerar como una generalización de la métrica K2:

$$g_{BD}(G:D) = log(p(G)) + \sum_{i=1}^{n} \left[ \sum_{j=1}^{q_i} \left[ log\left( \frac{\Gamma(\alpha_{ij})}{\Gamma(N_{ij}+\alpha_{ij})} \right) + \sum_{k=1}^{r_i} log\left( \frac{\Gamma(N_{ijk}+\alpha_{ijk})}{\Gamma(\alpha_{ijk})} \right) \right] \right]$$

donde $\alpha_{ijk}$ son los hiperparámetros para la distribución *a priori* de Dirichlet de los parámetros dada la estructura de red, y $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$. $\Gamma()$ es la distribución Gamma, si $c$ es un entero, entonces $\Gamma(c) = (c-1)!$. Obsérvese que si para todos los hiperparámetros usamos el valor $\alpha_{ijk} = 1$, obtenemos que la métrica K2 es un caso particular de la métrica BD.

En la práctica, la especificación de los hiperparámetros $\alpha_{ijk}$ es algo complicada (excepto si usamos asignaciones sin información, como se hace en K2). No obstante, considerando la asunción de equivalencia en verosimilitud [150], es posible especificar hiperparámetros de manera relativamente sencilla. El resultado es una métrica denominada BDe , cuya expresión es igual a la métrica BD pero los hiperparámetros se calculan de la siguiente forma:

$$\alpha_{ijk} = \alpha \times p(x_{ik}, \mathbf{w}_{ij}|G_0)$$

donde $p(.|G_0)$ representa la distribución de probabilidad asociada a la estructura inicial $G_0$ y $\alpha$ es un parámetro que representa el tamaño muestral equivalente.

Un caso particular de la métrica BDe, especialmente interesante, aparece cuando $p(x_{ik}, \mathbf{w}_{ij}|G_0) = \frac{1}{r_i q_i}$, es decir, cuando la red inicial asigna una probabilidad uniforme para cada configuración $\{X_i\} \cup Pa_g(X_i)$. La métrica resultante se denomina BDeu (BDe uniforme), que fue originalmente propuesta por [56]. Esta métrica depende de un sólo parámetro, el tamaño muestral equivalente $\alpha$, y se expresa como sigue:

$$g_{BDeu}(G:D) = log(p(G)) + \sum_{i=1}^{n} \left[ \sum_{j=1}^{q_i} \left[ log \left( \frac{\Gamma(\frac{\alpha}{q_i})}{\Gamma(N_{ij} + \frac{\alpha}{q_i})} \right) + \sum_{k=1}^{r_i} log \left( \frac{\Gamma(N_{ijk} + \frac{\alpha}{q_i})}{\Gamma(\frac{\alpha}{q_i})} \right) \right] \right]$$

Las métricas basadas en **teoría de la información** [49, 119, 204, 328, 329, 336, 78, 84, 154, 286] representan otra opción para medir el ajuste de un grafo dirigido acíclico al conjunto de datos. Están basadas en conceptos de teoría de la codificación e información.

En la codificación de un mensaje se trata de reducir lo más posible el número de elementos necesarios para representarlo atendiendo a su probabilidad de ocurrencia, esto es, los mensajes más frecuentes tienen códigos cortos y los mensajes menos frecuentes tendrán códigos largos. El principio de mínima longitud de descripción [288] (o MDL, del inglés *Minimum Description Length*), selecciona la codificación que conduce a una mínima longitud en la codificación de los mensajes. En el caso de las redes bayesianas, modelos muy complejos serán aquellos donde los nodos están densamente conectados (el caso extremo sería un grafo completo) y serán redes muy precisas, bastante ajustadas a los datos. No obstante, redes tan complejas suponen serios problemas de comprensión, computación y sobreajuste, por lo que se buscan redes más simples aunque sean menos precisas.

En nuestro caso, mediante redes bayesianas queremos representar el conjunto de datos $D$. Por tanto, la longitud de descripción incluye la longitud requerida para representar la red y, además, la longitud necesaria para representar los datos dada la red [49, 50, 119, 204, 328]. Para representar la red, debemos guardar su probabilidades, y esto requiere una longitud que es proporcional al número de parámetros libres de la factorización de la distribución de probabilidad conjunta. Este valor, denominado complejidad de la red $G$ y denotado como $C(G)$, es:

$$C(G) = \sum_{i=1}^{n} (r_i - 1) q_i$$

El factor de proporcionalidad usual es $\frac{1}{2}log(N)$ [289], esto es, la longitud media para almacenar un número de rango 0 a $N$. Por tanto, la longitud de descripción de la red es:

$$\frac{1}{2}C(G)log(N)$$

Observando la descripción de los datos dado el modelo, usando códigos de Hufmann su longitud resulta el opuesto del logaritmo de la verosimilitud si estimamos los parámetros con la frecuencia relativa. El logaritmo de la verosimilitud de los datos se puede ver como el logaritmo de la probabilidad de los datos dada la estructura. Se puede expresar de la siguiente manera [50]:

$$LL_D(G) = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk} log\frac{N_{ijk}}{N_{ij}}$$

Por tanto, la métrica MDL (cambiando los signos para tratar con un problema de maximización) quedaría como:

$$g_{MDL}(G:D) = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk} log\frac{N_{ijk}}{N_{ij}} - \frac{1}{2}C(Red)logN \qquad (1.3)$$

Otra forma de estimar la calidad de una red bayesiana es usar medidas basadas en teoría de la información y algunas de estas están íntimamente relacionadas con la anterior expresión. La idea básica consiste en buscar la estructura de red que mejor se ajusta a los datos, penalizada por el número de parámetros necesarios para especificar su correspondiente función de probabilidad conjunta. Esto nos lleva a una generalización de la métrica de la ecuación 1.3:

$$g(G:D) = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk} log\frac{N_{ijk}}{N_{ij}} - C(G)f(N)$$

donde $f(N)$ es una función de penalización no negativa. Si $f(N) = 1$, tendremos la métrica AIC (del inglés *Akaike Information Criterion*) [7], si $f(N) = \frac{1}{2}log(N)$ entonces tendremos la métrica BIC (del inglés, *Bayesian Information Criterion*) [302], que coincide con la métrica MDL. Si $f(N) = 0$, tenemos una métrica por máxima verosimilitud, aunque no es muy útil usar este último criterio, pues la mejor red usando esta métrica siempre es una red completa.

**Métodos de búsqueda.**

Como se ha visto, el aprendizaje de la estructura es un problema NP-difícil [75], por lo que no es posible hacer una búsqueda exhaustiva por el conjunto de todas las posibles estructuras con el objetivo de encontrar la mejor red. Es por ello que la búsqueda se hace de forma aproximada usando algoritmos de búsqueda heurísticos y probabilísticos.

Los algoritmos de búsqueda más usados suelen ser métodos de búsqueda local [56, 80, 76, 88, 150]. No obstante existen otros métodos basados en otras heurísticas de búsqueda, como la búsqueda tabú [50, 238] o ramificación y acotación [336].

También se han propuesto métodos de búsqueda probabilísticos como el enfriamiento simulado [76], algoritmos genéticos y programación evolutiva [209, 241, 361], métodos Monte Carlo en cadenas de Markov (abreviadamente MCMC, del inglés *Markov Chain Monte Carlo* [193, 241], búsquedas de entorno variable [90], colonias de hormigas [86], procedimientos de búsqueda voraces y aleatorios [87] o algoritmos de estimación de distribuciones [43].

#### 1.4.2.3. Aprendizaje automático: aproximaciones híbridas.

Existe una tercera aproximación que engloba tanto tests de independencia como búsquedas guiadas por métrica, sin pertenecer claramente a ninguno de los anteriores enfoques expuestos [4, 5, 317, 323, 83, 88]. Encontramos tres formas distintas en la hibridación de los dos enfoques.

Una primera, donde los dos enfoques se mantienen como procesos separados, y se entremezclan de diversas formas. Por ejemplo, el algoritmo CB [317], utiliza tests de independencia para obtener un grafo dirigido acíclico, utilizando el orden que hay en las variables de dicho grafo para usar el algoritmo K2, se itera el proceso aumentando en cada paso el orden permitido de los tests de independencia. En el algoritmo EGS [83], utiliza el algoritmo PC de forma iterativa, variando el nivel de confianza asociado a cada test y el orden en que se realizan los tests. Se evalúan los modelos resultantes con una métrica. Otra propuesta, es el algoritmo IMAPR [88] donde usa una búsqueda local con reinicios basados en relaciones de independencia.

Otra forma de hibridar ambos enfoques, es cuando se emplean tests de in-

dependencia para limitar/detener el proceso de búsqueda guiado por métrica. Por ejemplo, el algoritmo BENEDICT [5], se basa en cuantificar la discrepancia entre cualquier red candidata y la base de datos. Se miden las discrepancias entre las independencias condicionales representadas en la red (a través del concepto de d-separación, separación direccional o independencia gráfica) con las correspondientes independencias condicionales que puedan calcularse a partir de la base de datos. Estas discrepancias se usarán como métrica. El algoritmo parte de un grafo completamente inconexo, y en cada iteración se prueba a insertar cada uno de los arcos candidatos, eligiendo aquel que produce una mayor disminución de la discrepancia, e incluyéndolo en el grafo de forma permanente. Se eliminan arcos candidatos usando tests de independencia y se detiene la búsqueda cuando no quedan arcos candidatos.

La tercera manera de unir ambos enfoques, sería mezclar ideas provenientes de los métodos basados en tests de independencia para definir la métrica. Como representante de esta categoría, tenemos la métrica MIT [85], donde se combinan tests de independencia basados en la distribución chi cuadrado e información mutua, para medir el grado de interacción entre cada variable y sus padres. El resultado es una métrica descomponible basada en medidas de información y tests de independencia.

### 1.4.3. Aprendizaje automático de redes bayesianas usando información del experto.

En ciertos casos tenemos conjuntos de datos a partir de los cuales podemos construir nuestras redes bayesianas mediante aprendizaje automático pero además contamos con la ventaja de disponer de algún experto que nos guíe en el proceso. En otros casos no será fácil elicitar la información de un experto, por lo que las piezas de conocimiento que obtengamos del mismo serán las que usaremos.

Esta ayuda adicional puede ser bastante útil para restringir el espacio de búsqueda y, así, mejorar el rendimiento de los métodos de aprendizaje, de forma que encuentre mejores soluciones o lo haga en menor tiempo. Esta búsqueda guiada por un experto puede ser bastante valiosa en dominios muy

complejos.

No hay una forma unificada de incluir la información del experto, por lo que se presentará en el tercer capítulo una metodología para poder incluir información de un experto de forma eficiente y sencilla. No obstante, hay distintas aproximaciones que intentan utilizar la información del experto desde distintos ángulos:

- Información a priori: antes de comenzar el proceso de búsqueda se establece una distribución a priori [150] para que influya en el proceso de búsqueda, de forma que se refuercen aquellas relaciones de dependencia e independencia que el experto conozca.

- Estructura inicial: en algunos casos, en el aprendizaje estructural de la red, no se parte de un grafo vacío o un grafo completo sino que el punto de partida es un grafo facilitado por el experto [258]. O también, se puede obtener el grafo de la red bayesiana a partir de un experto y luego realizar un aprendizaje paramétrico [147].

- Espacio de búsqueda más adecuado para un dominio: puesto que no existe un método mejor que otro para todos los problemas [360], un experto podría recomendarnos la topología de red más apropiada para un problema concreto. Por ejemplo, el clasificador naïve bayes (que se presentará en la sección 1.5.4.1) funciona muy bien en dominios médicos donde los síntomas son vistos como independientes entre sí dada la enfermedad [205].

- Ordenación de variables: en algunos algoritmos, como es el caso del método de búsqueda basado en la métrica K2 [80], una correcta ordenación de las variables asegura un correcto funcionamiento del método. Si obtenemos un orden, al menos parcial, de las mismas facilitado por un experto, obtendremos mejores resultados [317]. El orden de las variables debería ser compatible con las relaciones causa-efecto.

## 1.5. Redes bayesianas usadas como clasificadores.

Dentro del aprendizaje automático se, conoce como *clasificación* a la tarea de asociar los valores que toma un conjunto de variables (también denominadas *características*) con un conjunto discreto de valores, denominados *clases*. La idea es construir de forma automática un modelo, denominado *clasificador*, a partir de un conjunto de datos, y luego, utilizarlo para asignar clases a nuevos datos que no han sido usados en la construcción del modelo.

El funcionamiento de un clasificador depende en gran medida de las propiedades de los datos que queramos clasificar, por lo que no hay un tipo de clasificador que sea mejor que el resto de metodologías para todos los problemas [360].

Dentro del aprendizaje automático para clasificación podemos distinguir entre *clasificación no supervisada* y *clasificación supervisada*. Se diferencian en que el aprendizaje supervisado utiliza un conjunto de muestras previamente clasificadas, es decir, que para cada caso de la muestra conteniendo valores de las variables tiene asociada una clase. En clasificación no supervisada, por el contrario, se tiene el conjunto de datos sin ningún conocimiento *a priori* de su clase: no se conoce la clase de cada caso o incluso, en la mayoría de los casos, se desconoce cuantas clases hay.

### 1.5.1. Clasificación supervisada y no supervisada.

En *clasificación supervisada* el conjunto de variables a considerar será $\mathbf{V} = \mathbf{X} \cup \{C\}$. Donde $C$ es la variable a clasificar (también denominada variable distinguida o, simplemente, clase) y las variables en $\mathbf{X}$ serán usadas para predecir los valores de $C$ y se les denomina variables predictoras, variables inductoras, características o atributos. El objetivo es describir la variable clase en función de los atributos y poder calcular la probabilidad condicionada de la clase dada cualquier configuración de los atributos, $p(C|\mathbf{X})$. Las redes bayesianas usadas en tareas de clasificación supervisada se denominan genéricamente, *clasificadores bayesianos*.

En *clasificación no supervisada* (o agrupamiento -*clustering*-), partiendo de un conjunto de variables $\mathbf{V} = \mathbf{X}$ se trata de asignar cada caso a un grupo (*clúster*) $C$, es decir, el objetivo es descubrir una estructura de clases en los

datos, de tal manera que los casos que pertenezcan a una misma clase o grupo presenten una gran homogeneidad, mientras que los casos que pertenezcan a distintos agrupamientos o clasificaciones deben ser muy heterogéneos entre sí. Obsérvese, que se puede ver la clasificación no supervisada, como una clasificación supervisada donde todos los valores de la variable clase son desconocidos, es decir, como aprendizaje de datos incompletos. Dentro de la clasificación no supervisada hay distintos métodos de aprendizaje, usando redes bayesianas, a partir de datos incompletos [67, 115, 335, 231].

### 1.5.2. Evaluación de clasificadores.

Un detalle importante cuando construimos clasificadores es cuantificar de alguna manera lo buenos o malos que son [350]. Por ejemplo, si usamos un clasificador para detección de cáncer no nos podemos permitir el lujo de que falle tres de cada cuatro pacientes. A la hora de evaluar un clasificador se hará teniendo en cuenta distintos criterios, como puede ser el tiempo que tardamos en construirlo, la interpretabilidad del modelo obtenido, la sencillez del modelo (cuanto más sencillo, mayor capacidad de abstracción) o diferencias respecto al modelo original; pero al que más atención se le presta es a la *precisión* del clasificador (o, a la inversa, la tasa de error) que posee.

La precisión de un clasificador es la probabilidad con la que clasifica correctamente un caso seleccionado al azar [196], o también, lo podemos ver como el número de casos clasificados correctamente entre el número total de elementos.

$$precisión = \frac{número\,de\,aciertos}{número\,de\,casos}$$

Además de ser la medida más aceptada para la evaluación de un clasificador, la precisión es utilizada en algunos procedimientos para guiar la construcción del clasificador, por ello vamos a exponer distintas formas de obtener su valor:

- *Estimación por resustitución (resubstitution estimate) o error aparente*: este modo, el más simple, consiste en utilizar los mismos datos que se han utilizado para construir el clasificador para ver cuántos predice correctamente.

- *Holdout*: se basa en partir el conjunto de datos aleatoriamente en dos grupos. El denominado conjunto de entrenamiento (normalmente 2/3 del número total de casos) con el cual se construye el clasificador y el conjunto de testeo o validación (construido con el 1/3 restante) usado para estimar la precisión del clasificador.

- *Remuestreo (random subsampling)*: es una variación del sistema anterior donde se realizan diferentes particiones de los conjuntos de entrenamiento y test. Obteniéndose la precisión del clasificador a partir de la media obtenida en los distintos conjuntos de test.

- *Validación cruzada de k-hojas(k-fold cross-validation)* [326]: se puede ver como una generalización del criterio de remuestreo. Hacemos $k$ particiones del conjunto de datos mutuamente excluyentes y de igual tamaño. $k-1$ conjuntos se utilizan para construir el clasificador y se valida con el conjunto restante. Este paso se efectúa $k$ veces y la estimación de la precisión del clasificador se obtiene como la media de las $k$ mediciones realizadas.

- *Dejar-uno-fuera (leave-one-out)* [203]: es un caso particular de la validación cruzada, donde se hacen tantas particiones como casos tenga el conjunto de datos. De esta forma los conjuntos de validación tienen un sólo caso y los de entrenamiento todos los casos menos ese en particular. Al tener que construir el clasificador tantas veces como casos tenga el conjunto de datos se hace bastante costoso en tiempo, a no ser que se pueda obtener una fórmular cerrada para el error.

- *Bootstrapping* [102]: para evaluar un clasificador de forma efectiva cuando se tienen conjuntos de datos con pocas muestras se suele utilizar el método de dejar-uno-fuera. No obstante, con conjuntos pequeños de datos, suele mostrar una varianza alta. El bootstrapping (o bootstrap) es un método de remuestreo propuesto por Bradley Efron en 1979 [101], donde para una muestra de tamaño $n$ se genera el conjunto de entrenamiento con $n$ casos mediante muestreo con reemplazamiento, es decir, cogemos un caso de forma aleatoria del conjunto de datos para el conjunto de entrenamiento y lo volvemos a dejar en el conjunto de datos, de esta forma puede haber casos repetidos en el conjunto de entrenamiento. El conjunto de test se genera cogiendo aquellos casos que no estén en el conjunto de entrenamiento. Este proceso (crear el conjunto de

entrenamiento mediante muestreo con reemplazo) se realiza un número elevado de veces. La media de las distintas precisiones calculadas sirve como estimación de la precisión verdadera.

Algunas veces es interesante no sólo conocer la precisión del clasificador o la tasa de error, sino que es importante el sentido en el que se equivoca. Pongamos un ejemplo extremo: estamos diagnosticando mediante un clasificador si un paciente tiene o no cáncer; si el clasificador se equivoca y toma a una persona sana como enferma (ésto se conoce como *falso positivo*), es un error que probablemente se descubrirá más tarde en posteriores análisis (en el peor de los casos se habrá generado una serie de pruebas y preocupaciones innecesarias). Por el contrario, si el clasificador toma a una persona como sana cuando en realidad está enfermo (lo que se conoce como un *falso negativo*), quizás a la hora de solucionar el error ya sea demasiado tarde (perdemos la posibilidad de tratar al paciente a tiempo).

Cuando distinguir entre los distintos tipos de errores es importante, entonces se puede usar una *matriz de confusión* (también llamada *tabla de contingencia*) para mostrar los diferentes tipos de error. Si tenemos un problema con dos clases (por ejemplo, sano y enfermo), como se puede ver en la tabla 1.1, un clasificador puede dar la siguiente salida para un nuevo caso: verdadero positivo, predice que el paciente tiene la enfermedad y es verdad, verdadero negativo si acierta que el paciente no está enfermo, falso positivo pronostica enfermo estando el paciente sano y, finalmente, falso negativo, predice sano pero el paciente está enfermo.

| | Enfermo | Sano |
|---|---|---|
| Enfermo | Verdadero positivo | Falso positivo |
| Sano | Falso negativo | Verdadero negativo |

Tabla 1.1: Matriz de confusión para un problema con dos clases.

A partir de la matriz de confusión podemos construir algunas medidas que nos serán de utilidad. La *sensibilidad* es la probabilidad de clasificar correctamente a un individuo enfermo, por tanto, es la capacidad del clasificador para detectar la clase positiva (la que más interesa, en el ejemplo, que

está enfermo), se define como:

$$sensibilidad = \frac{verdaderos\_positivos}{verdaderos\_positivos + falsos\_negativos}$$

por otro lado, definimos la *especificidad* como la probabilidad de clasificar correctamente a un individuo sano, en otras palabras, se puede definir la especificidad como la capacidad de clasificar correctamente a un individuo cuyo estado real sea negativo para la prueba que se hace (en nuestro ejemplo, que esté sano). Se puede calcular a partir de la matriz de confusión como:

$$especificidad = \frac{verdaderos\_negativos}{verdadero\_negativos + falsos\_positivos}$$

En problemas donde distinguir el tipo de error es importante, se pueden utilizar las *curvas ROC* (del inglés, *Receiver Operating Characteristics*) [110]. Las curvas ROC aunque tienen su origen en la detección de señales de radar, se usan habitualmente en la toma de decisiones médicas [330] y además nos van a permitir evaluar de forma gráfica el funcionamiento de un clasificador. Son curvas en las que se presenta la sensibilidad en función de los falsos positivos (complementario de la especificidad) para distintos resultados de un clasificador.

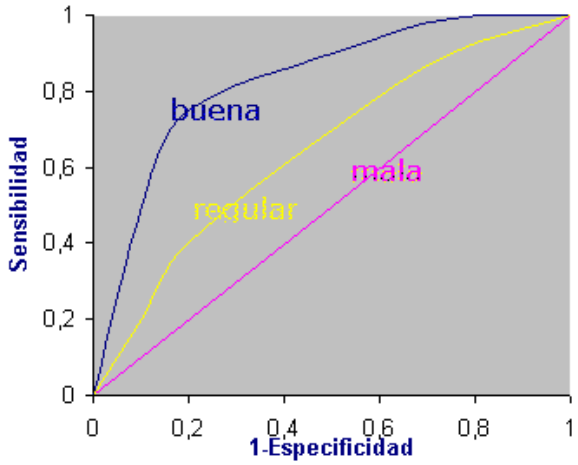


Figura 1.5: Tipos de curvas ROC.

Cuanto mayor sea el área bajo la curva ROC, mejor será el clasificador. La mejor predicción posible sería un método que pasara por la esquina superior izquierda, ya que representaría un 100 % de sensibilidad (no hay falsos negativos) y un 100 % de especificidad (no hay falsos positivos). Por tanto, cuanto más distante está la curva de la diagonal mejor será el clasificador, como se puede ver en la figura 1.5.

### 1.5.3. Enfoques de filtrado y de envoltura en clasificadores.

La construcción de un clasificador, en algunos casos, conlleva un proceso de búsqueda en la estructura del mismo, con el objetivo de maximizar la precisión del clasificador final y así, obtener un mejor modelo. Esa búsqueda de la estructura de los clasificadores pueden hacerse en base a dos enfoques:

- Filtrado (*filter*): se utiliza una medida (por ejemplo, la verosimilitud, la entropía o la información mutua) para medir la bondad de las distintas estructuras candidatas en el proceso de búsqueda. Esta medida es independiente del clasificador utilizado.

- Envoltura (*wrapper*): se realiza una búsqueda guiada por la precisión del clasificador con el que se está trabajando, como si fuera una caja negra que evalúa cada una de las estructuras candidatas del espacio de búsqueda. Normalmente se utiliza una validación cruzada sobre cada una de las estructuras candidatas, por lo que es computacionalmente un método bastante costoso.

Los métodos de envoltura presentan un mejor comportamiento en la precisión de los clasificadores que los métodos de filtrado, algo lógico pues optimizan la precisión del clasificador en su proceso de búsqueda. No obstante los clasificadores de filtrado suelen presentar un comportamiento más robusto frente a un sobreajuste a los datos y son más eficientes.

### 1.5.4. Clasificadores bayesianos.

En este trabajo nos vamos a centrar en el uso de clasificadores bayesianos, es decir, redes bayesianas utilizadas para tareas de clasificación supervisada

construidas mediante aprendizaje automático. Cualquier red bayesiana puede ser utilizada como un clasificador. Téngase en cuenta que el manto de Markov de la variable clase en la red aprendida, puede utilizarse para hacer predicciones.

El primer clasificador bayesiano que vamos a considerar es el clasificador *naïve bayes* (NB) [100, 206]. Este clasificador se basa en dos supuestos:

- Cada atributo es condicionalmente independiente de los otros atributos dada la clase
- Todos los atributos tienen influencia sobre la clase.

En la representación gráfica del NB (ver figura 1.6) tenemos todos los arcos dirigidos desde la clase hacia los atributos. El éxito de este clasificador reside en su simplicidad y a que muestra un comportamiento extremadamente competitivo en la precisión de sus predicciones. Mejora a muchos otros clasificadores más sofisticados en un amplio espectro de bases de datos (especialmente en aquellos casos en los que los atributos no están fuertemente correlados) [206, 118]. No obstante, estas dos restricciones que impone el NB presentan problemas cuando los atributos no son condicionalmente independientes entre sí (debido a la primera restricción) [260] o cuando hay variables redundantes o irrelevantes (debido a la segunda restricción) [205, 176].

El relativo éxito del clasificador NB ha motivado el desarrollo de otros métodos basados en intentar mejorarlo relajando algunas de las dos hipótesis que hace. Una forma es, empezando desde la topología del NB, completarlo mediante la adición de arcos entre atributos, es decir, restringiendo la estructura del grafo. La idea es relajar la restricción de que todos los atributos son condicionalmente independientes entre sí. Por ejemplo, en el clasificador TAN (*tree-augmented Naïve Bayesian network*) [118], los atributos se enlazan siguiendo la estructura de un árbol dirigido. En el clasificador BAN (*Bayesian network augmented Naïve Bayesian classifier*), se fija la estructura NB y se usa un algoritmo de aprendizaje de redes bayesianas, pero sólo en los atributos [69].

Hay otras variaciones al modelo NB que intentan superar la restricción de que los atributos sean independientes entre sí pero usando otro enfoque. Este es el caso del *clasificador semi-naïve bayes* [201] y el modelo propuesto

por [260], que comparten la idea de mezclar aquellos atributos correlados en nuevos atributos compuestos.

Respecto a la segunda restricción, que todos los atributos influyen en la variable clase, la forma de mejorar el clasificador NB es quitar los atributos correlacionados y los irrelevantes usando *selección de características (feature subset selection -FSS-)*. Como exponente de este enfoque tenemos el *clasificador naïve bayes selectivo* [205, 176] que busca un buen subconjunto de atributos, empezando por un conjunto vacío e iterativamente va introduciendo aquel atributo que mejore más las predicciones del clasificador.

Obsérvese que los clasificadores expuestos tienen su estructura restringida en cierto modo, pues tienen en común que todas los atributos son hijos de la variable a clasificar. No obstante y pese a su peor comportamiento como clasificadores, también se han utilizado en la literatura redes bayesianas sin ningún tipo de restricción en tareas de clasificación [118, 69, 70].

Asimismo podemos considerar a las multiredes bayesianas [128] como otro tipo de clasificador bayesiano que extiende el formalismo de las redes bayesianas. Las multiredes bayesianas nos permiten representar independencias dependientes del contexto. En este modelo tenemos distintas redes bayesianas para distintos valores de la clase, o bien, distintas redes bayesianas para los distintos valores de un atributo.

A continuación vamos a ver con más profundidad los clasificadores bayesianos más usados en la literatura.

#### 1.5.4.1. Naïve bayes.

Como hemos visto en la introducción, el naïve bayes [100] es uno de los modelos más simples y más utilizados. Su nombre tiene el origen en la hipótesis ingenua (naïve) de que los atributos son condicionalmente independientes dada la variable a clasificar (véase la figura 1.6). Dicha hipótesis tiene una serie de implicaciones geométricas, que han sido estudiadas en [272].

La probabilidad de que el $j-$ésimo ejemplo pertenezca a la clase $i$-ésima de la variable a clasificar $C$, puede calcularse, sin más que aplicar el teorema

de Bayes, de la siguiente manera:

$$P(C = c_i|X_1 = x_{1j}, ..., X_p = x_{pj}) \propto P(C = c_i) \times P(X_1 = x_{1j}, ..., X_p = x_{pj}|C = c_i)$$

En el caso de que las variables predictoras sean condicionalmente independientes dada la variable $C$, se obtiene que:

$$P(C = c_i|X_1 = x_{1j}, ..., X_p = x_{pj}) \propto P(C = c_i) \times \prod_{r=1}^{p} P(X_r = x_{rj}|C = c_i)$$

Aunque la simplicidad del naïve bayes y la restricción de que los atributos sean independientes entre si, juega en contra de este modelo, la literatura existente [205] muestra el alto grado de aciertos que consigue en muchos dominios, especialmente en los relacionados con la medicina. Muchos investigadores piensan que el naïve bayes está basado en la idea de que los médicos, al realizar un diagnóstico, recogen los atributos al igual que los utiliza el naïve bayes para clasificar, es decir, independientes dada la clase.

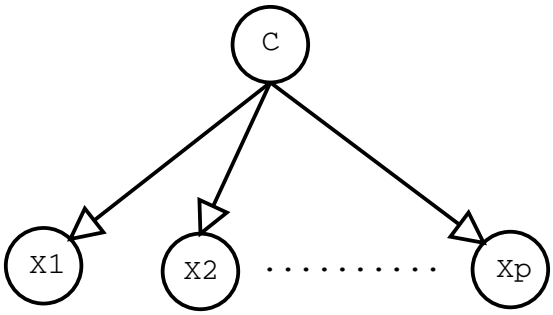


Figura 1.6: Estructura del clasificador naïve bayes.

#### 1.5.4.2. Estructura de árbol aumentado (TAN).

Para superar la restricción de que los atributos son independientes entre sí, algo que no es cierto cuando los atributos están fuertemente correlacionados, Friedman y Col. [118] presentan un método de construcción de lo que denominan estructuras TAN (*Tree Augmented Naïve Bayes*), que obtiene mejores resultados que los obtenidos por el naïve bayes, a la vez que no deteriora de manera significativa la simplicidad computacional y la robustez del anterior.

Podemos decir que un modelo TAN es una red bayesiana donde el conjunto de padres de la variable a clasificar, $C$, es vacío, mientras que el conjunto de variables padres de cada una de las variables predictoras, $X_i$, contiene necesariamente a la variable a clasificar, y como mucho otra variable. Véase por ejemplo la figura 1.7.

Los anteriores autores proponen un algoritmo -adaptación de [78]- que utiliza el concepto de *información mutua* entre variables predictoras condicionada a la variable a clasificar. La función se define como:

$$I_P(X;Y|C) = \sum_{x,y,c} logP(x,y,c)\frac{P(x,y|c)}{P(x|c)P(y|c)}$$

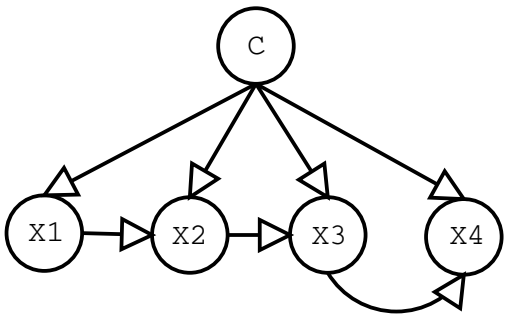


Figura 1.7: Estructura de árbol aumentado.

De manera simple, podemos decir que la función anterior mide la información que la variable $Y$ proporciona sobre la variable $X$ cuando el valor de $C$ es conocido.

El algoritmo propuesto por Friedman y col. [118], garantiza que la estructura TAN obtenida tiene asociada la máxima verosimilitud entre todas las posibles estructuras de TAN (ver Algoritmo 1).

**Algoritmo 1** Algoritmo de filtrado para construir un clasificador TAN

1: Calcular $I_p(X_i, X_j|C)$ para cada par de variables predictoras con $i \neq j$.
2: Construir un grafo no dirigido completo en el cual los vértices son las variables predictoras $X_1, ..., X_n$. Asignar a cada arista conectando las variables $X_i$ y $X_j$ un peso dado por $I_p(X_i; X_j|C)$.
3: Construir un árbol expandido de máximo peso.
4: Transformar el árbol resultante no dirigido en uno dirigido, escogiendo una variable raíz, y direccionando todas las aristas partiendo del nodo raíz.
5: Construir un modelo TAN añadiendo un nodo etiquetado como $C$, y posteriormente un arco desde $C$ a cada variable predictora $X_i$.

También se han propuesto enfoques de envoltura para el clasificador TAN [181, 273]. En este enfoque se parte de un TAN con dos variables y luego se van añadiendo atributos o arcos que verifiquen la estructura de árbol, tal y como se muestra en el Algoritmo 2.

**Algoritmo 2** Algoritmo de envoltura para construir un clasificador TAN

1: Inicializar el conjunto de variables del clasificador TAN **S** como vacío.
2: Seleccionar las dos variables con más precisión y añadirlas a **S**.
3: Repetir hasta que no haya mejora, la mejor opción entre:

1. Considerar cada variable no incluida en **S**, que sea condicionalmente independiente a las variables ya incluidas en **S** dada la clase.

2. Considerar cada posible arco entre cada par de atributos de **S** que no invalide la estructura TAN.

Otro método, variante del clasificador TAN, es la estructura de bosque aumentado (*forest-augmented naïve Bayesian network -FAN-*) [222], donde las dependencias entre los atributos están restringidas a forma de bosque (un conjunto de árboles); es decir, una estructura donde podemos tener más de un nodo raíz, pero a lo sumo un sólo padre por nodo además de la clase.

#### 1.5.4.3. Clasificadores bayesianos k-dependientes (KDB).

Como evolución del clasificador TAN y también con el objetivo de superar la restricción de que los atributos sean condicionalmente independientes en-

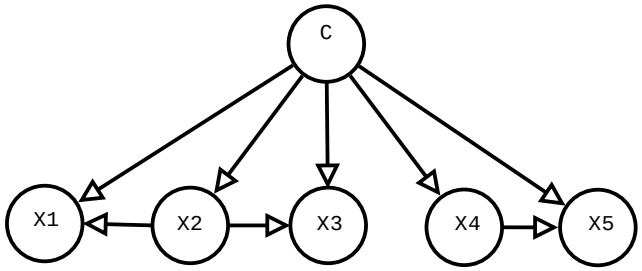


Figura 1.8: Estructura de bosque aumentado.

tre sí, Sahami [296] generaliza el modelo TAN mediante los *clasificadores bayesianos k-dependientes*.

Se define un clasificador bayesiano k-dependiente como una red bayesiana que contiene la estructura de un clasificador naïve bayes y permite a cada atributo $X_i$ tener un máximo de $k$ atributos como nodos padre. En otras palabras, $Pa(X_i) = \{C, \mathbf{X_{pa_i}}\}$ donde $\mathbf{X_{pa_i}}$ es un conjunto de como máximo $k$ atributos , y $Pa(C) = \emptyset$.

De esta forma podemos ver que el naïve bayes y el naïve bayes selectivo son clasificadores bayesianos 0-dependientes. Mientras que el TAN (o el FAN) se puede considerar un clasificador bayesiano 1-dependiente. Variando el valor de k nos podemos mover por el espectro de dependencias de variables inductoras. Podemos verlo gráficamente en las figuras 1.6 (clasificador 0-dependiente), 1.7 y 1.9.

En [296] se proporciona un algoritmo para construir clasificadores bayesianos k-dependientes, de forma que en función de la capacidad computacional disponible podemos elegir un grado, k, más alto o bajo de dependencias (ver Algoritmo 3).

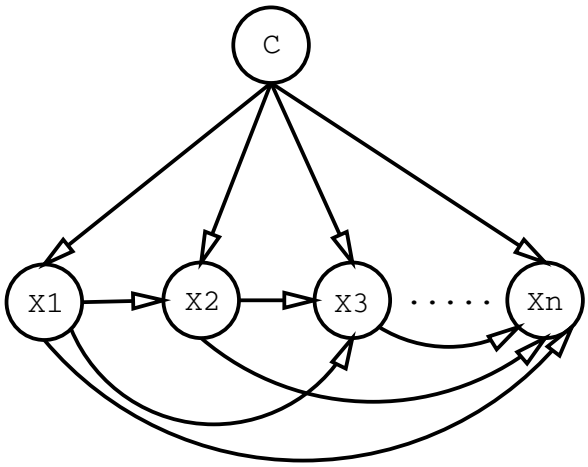


Figura 1.9: Clasificador bayesiano k-dependiente.

**Algoritmo 3** Algoritmo KDB (*K-Dependence Bayesian Classifier*)

1: Para cada atributo $X_i$, se calcula la información mutua $I(X_i; C)$, con la variable clase.
2: Se calcula la información mutua condicionada a la clase $I(X_i; X_j|C)$para cada par de atributos $X_i$ y $X_j$, donde $i \neq j$
3: Inicialicemos a vacía la lista $S$ de variables usadas.
4: Se empieza a construir la red bayesiana $BN$ con un sólo nodo, la clase $C$
5: **mientras** $S$ no incluya todos los atributos **repite**
6: Seleccionar el atributo $X_{max}$ que no esté en $S$ y que tenga el mayor valor de $I(X_{max}, C)$
7: Añadir un nodo a la $BN$ representando a $X_{max}$
8: Añadir un arco desde $C$ a $X_{max}$ en la $BN$
9: Añadir $m = min(|S|, k)$ arcos desde $m$ atributos distintos $X_j$ en $S$ hasta $X_{max}$ con el valor más alto de $I(X_{max}; X_j|C)$
10: Añadir $X_{max}$ a $S$
11: **fin mientras**
12: Calcular la tablas de probabilidad condicionada inferidas por la estructura del $BN$ a partir de la lista de casos.

Hay una variación de este algoritmo denominado KDB-$\theta$, donde en el paso 9 se tiene en cuenta los $m$ atributos distintos $X_j$ en $S$ con el valor más alto

de $I(X_{max}; X_j|C)$ y sólo se añaden arcos de $X_j$ a $X_{max}$ si $I(X_{max}; X_j|C) > \theta$, donde $\theta$ es un umbral de información mutua, normalmente $\theta = 0{,}003$.

#### 1.5.4.4. Naïve bayes aumentado a red bayesiana (BAN).

El siguiente paso en la evolución de los clasificadores bayesianos sería el naïve bayes aumentado a red bayesiana (Bayesian network augmented naïve Bayesian classifier -BAN-) [69]. Partiendo de la estructura simple del naïve bayes, hemos pasado a la estructura de árbol que posee el TAN, y después a la estructura que posee un clasificador bayesiano k-dependiente. De esta forma se relaja completamente la restricción que impone que los atributos sean independientes entre sí dada la clase. Obsérvese también que un clasificador BAN con $n$ atributos se puede considerar un clasificador k-dependiente con $k = n$.

Más formalmente, sean $X_1, X_2, ..., X_n$ las variables predictoras y sea $C$ la variable a clasificar. El Algoritmo 4 muestra un esquema general de aprendizaje de un clasificador BAN.

**Algoritmo 4** Algoritmo para la construcción de un clasificador BAN

1: Nos quedamos sólo con las variables predictoras $X_1, X_2, ..., X_n$ y sus correspondientes casos del conjunto de entrenamiento
2: Construimos una red bayesiana con dichas variables utilizando algún algoritmo de aprendizaje estructural de redes bayesianas (no de clasificadores) de los que hay en la literatura.
3: Añadimos $C$ como nodo padre de todas las variables $X_i$ de la red.
4: Hacemos el aprendizaje paramétrico de la estructura obtenida.

#### 1.5.4.5. Semi naïve bayes.

Existen otros métodos que tienen la intención de mejorar la fuerte condición de independencia entre las variables predictoras que posee el clasificador naïve bayes, sin complicar la estructura existente entre dichas variables, como hemos visto en los modelos anteriores. En este caso nos encontramos los modelos propuestos por Kononenko [201] y Pazzani [260], que comparten la misma filosofía de mezclar distintos atributos correlacionados en nuevos atributos compuestos. Como en el clasificador naïve bayes, los atributos compuestos se consideran independientes entre ellos dada la clase, pero no

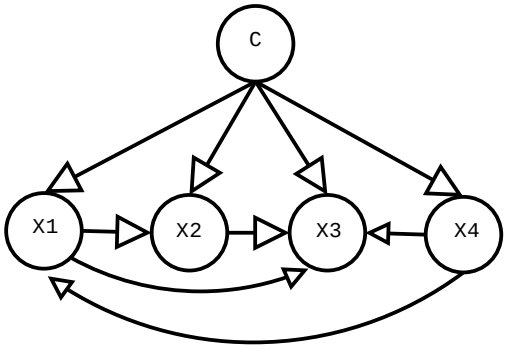


Figura 1.10: Naïve bayes aumentado a red bayesiana.

se consideran independientes los atributos de un mismo atributo compuesto. A éste tipo de clasificadores bayesianos se les denomina comúnmente semi naïve bayes.

El modelo propuesto por Kononenko en [201] hace el producto cartesiano de los valores de aquellas variables que cumplan una condición, eliminándose las variables originales. Dicha condición aparece al mezclar el concepto de independencia y el concepto de la fiabilidad en la estimación de las probabilidades condicionales.

Pazzani [260] presenta un modelo que puede considerarse que se posiciona en un lugar intermedio entre los modelos extremos, en los que, por un parte se tienen que calcular las $(2^p - 1)r$ distribuciones de probabilidad - para el caso en que la variables $C$ admita $r$ posibles valores, y las variables predictoras sean dicotómicas-. Es decir, el modelo necesita las siguientes probabilidades:

$$P(X_1 = x_{1j}, ..., X_p = x_{pj}|C = c_i)$$

Por otra parte en el modelo naïve bayes, se hace necesario calcular:

$$P(C = c_i|X_1 = x_{1j}, ..., X_p = x_{pj}) \propto P(C = c_i) \times \prod_{r=1}^{p} P(X_r = x_{rj})$$

,

y por tanto no necesitaríamos más que $(r - 1)p$ probabilidades.

Veamos lo propuesto por Pazzani, siguiendo un ejemplo. Supongamos un dominio con 4 variables predictoras $X_1, X_2, X_3, X_4$ y una variable a predecir $C$. Supongamos asimismo que la variable $X_2$ no es relevante para $C$, y que

además las variables $X_1$ y $X_3$ son condicionalmente dependientes dada C. Tendríamos una situación que gráficamente puede ser expresada como se ve en la Figura 1.11.

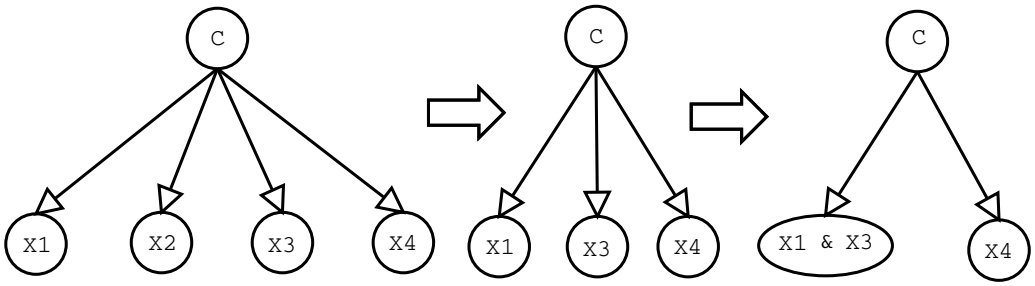


Figura 1.11: Semi naïve bayes.

A nivel de fórmulas lo expresaríamos de la siguiente forma:

$$P(C = c_i | X_1 = x_{1j}, ..., X_4 = x_{4j}) \propto$$

$$P(C = c_i) \times P((X_1 = x_{1j}, X_3 = x_{3j}) | C = c_i) \times P(C_4 = x_{4j} | C = c_i)$$

Lo que queda por determinar es qué variables son no relevantes, y por otra parte qué variables van a agruparse y necesitan que se estimen para las mismas las probabilidades condicionadas correspondientes.

Pazzani propone para la selección del modelo dos algoritmos voraces, siguiendo la filosofía estadística de modelización hacia adelante (algoritmo FSSJ -Algoritmo 5-) y modelización hacia atrás (algoritmo BSEJ -Algoritmo 6-).

**Algoritmo 5** Algoritmo FSSJ (Forward Sequential Selection and Joining)

1: Inicializar el conjunto de variables a utilizar a vacío. Clasificar todos los ejemplos en la clase más frecuente.
2: Repetir en cada paso la mejor opción entre:

   1. Considerar cada variable no usada como una nueva variable a incluir en el modelo, condicionalmente independiente de las variables ya incluidas dada la variable a clasificar

   2. Juntar cada variable no utilizada con una variable ya incluida en el clasificador.

3: Hasta que ninguna operación produzca mejoras

**Algoritmo 6** Algoritmo BSEJ (Backward Sequential Elimination and Joining)

1: Crear un clasificador bayesiano considerando todos los atributos como condicionalmente independientes dada la variable a clasificar.
2: Repetir en cada paso la mejor opción entre:

   1. Considerar el hecho de sustituir cada par de atributos usados por el clasificador con un nuevo atributo que sea la unión de dicho par de atributos.

   2. Considerar el hecho de eliminar cada atributo usado por el clasificador.

3: Hasta que ninguna operación produzca mejoras

#### 1.5.4.6. Naïve bayes selectivo.

Otra forma de mejorar el clasificador NB es relajando la asunción de que todos los atributos tienen influencia sobre la clase. Bajo este enfoque se encuentra el clasificador naïve bayes selectivo (en inglés, *Selective Naïve Bayes*) [205, 176], donde se construye el clasificador naïve bayes pero sólo utilizando aquellos atributos que son relevantes. Es decir, hace una selección de aquellos atributos que mejoran la capacidad del clasificador naïve bayes,

eliminando aquellos que sean irrelevantes y/o redundantes y que merman su eficacia.

Este clasificador se construye empezando con una lista vacía de atributos y se van añadiendo atributos mientras el clasificador vaya mejorando. La construcción del modelo termina cuando la inclusión de nuevas variables no produce ninguna mejora. En realidad, lo que se hace en este clasificador es una selección de variables (ver apartado 1.6) previa a la construcción del clasificador naïve bayes.

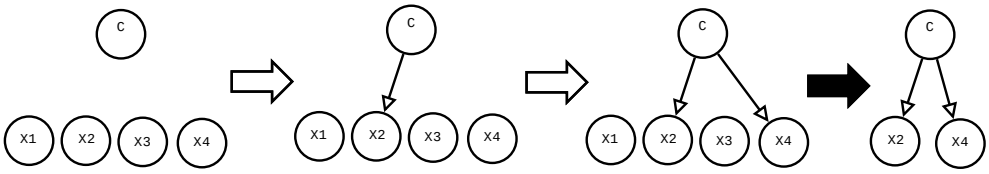


Figura 1.12: Naïve bayes selectivo.

#### 1.5.4.7. Clasificadores bayesianos sin restricciones estructurales.

Hasta el momento todos los clasificadores bayesianos que hemos visto tienen una restricción estructural común y es que la variable a clasificar es un nodo padre del resto de variables usadas en el clasificador. El siguiente paso sería eliminar dicha restricción, o mejor aún, quitar cualquier tipo de restricción acerca de la estructura y poder usar cualquier red bayesiana como un clasificador.

Si tenemos en cuenta que una variable es condicionalmente independiente del resto de variables dado su manto de Markov (variables padres, variables hijas y variables padre de las hijas), bastaría con quedarnos con el manto de Markov de la variable a clasificar (obsérvese que ninguna variable fuera del manto de Markov de la clase afecta a dicha clase). De esta forma podemos obtener un clasificador a partir de cualquier red bayesiana y además se hace una selección de características (ver apartado 1.6) al quedarnos con sólo aquellos atributos que influyen en la variable a clasificar. En la literatura existente, a este tipo de clasificadores se les ha llamado redes bayesianas sin restricciones (Unrestricted Bayesian Networks -UBN-) [118] o redes bayesianas generales (General Bayesian Network -GBN-) [69].

Obsérvese que este enfoque supera las dos restricciones del NB, puesto que los atributos no tienen por que ser independientes entre sí y además no todos los atributos influyen en la variable clase (sólo influyen aquellos que pertenecen al manto de Markov de la clase).

Por tanto a la hora de usar una red bayesiana cualquiera como clasificador, tenemos que seguir lo pasos que se describen en el Algoritmo 7.

**Algoritmo 7** Algoritmo para usar una red bayesiana como clasificador

1: Crear la estructura de la red bayesiana de forma manual, o bien, de forma automática con cualquiera de los algoritmos que existen en la literatura.
2: Una vez obtenida la estructura completa, nos quedamos sólo con el manto de Markov de la variable a clasificar.
3: Finalmente, hacemos un aprendizaje paramétrico para la estructura obtenida.

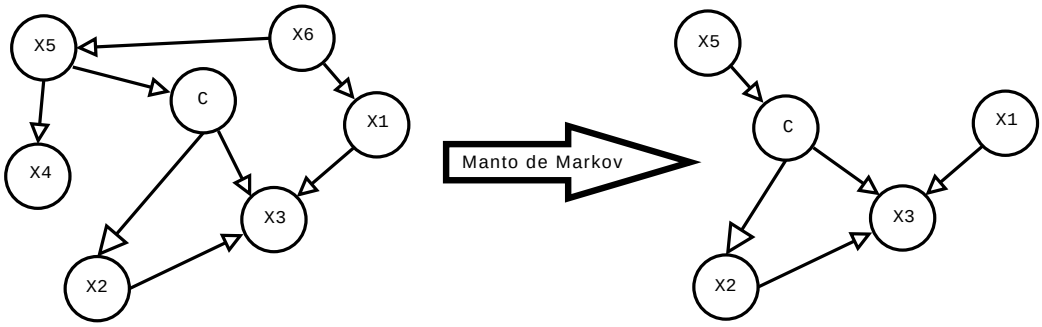


Figura 1.13: Red bayesiana como clasificador.

Como ventaja este clasificador bayesiano sin ningún tipo de restricción estructural nos permite tener una mayor capacidad de expresión que con un modelo más restringido. Como contrapartida, Friedman y col. demostraron en [118] que un método de búsqueda basado en una métrica de mínima longitud de descripción genera redes que maximizan dicha métrica pero que en realidad son pobres clasificadores; por lo tanto una red bayesiana construida de esta forma no se comporta bien como clasificador. Sin embargo, puede que estos experimentos tengan un fallo en el diseño, ya que la estimación paramétrica se realiza por máxima verosimilitud, sin ningún tipo de corrección para muestras pequeñas, por lo que las conclusiones a las que llegaron pueden no estar

soportadas. Así, Cheng y Greiner en [69] y [70] vieron que una red bayesiana construida mediante un algoritmo basado en tests de independencias no sufría de estas desventajas, obteniendo así un clasificador efectivo basado en una red bayesiana sin restricciones estructurales.

Otro tipo de clasificador íntimamente relacionado es el propuesto por Sierra y Larrañaga [315] donde se realiza una búsqueda sobre un espacio reducido de los posibles mantos de Markov de la variable a clasificar.

#### 1.5.4.8. Multiredes bayesianas.

Si observamos alguna de las propuestas anteriores descubriremos que la relación entre las variables es siempre la misma en cada una de las clases. Una mejora del modelo anterior sería considerar un clasificador distinto para cada uno de los valores de la clase. El formalismo que realiza esta mejora son las *multiredes bayesianas* [128] (ver figura 5.3). Las multiredes bayesianas son una extensión de las redes bayesianas ya que además nos permiten representar independencias asimétricas [147].

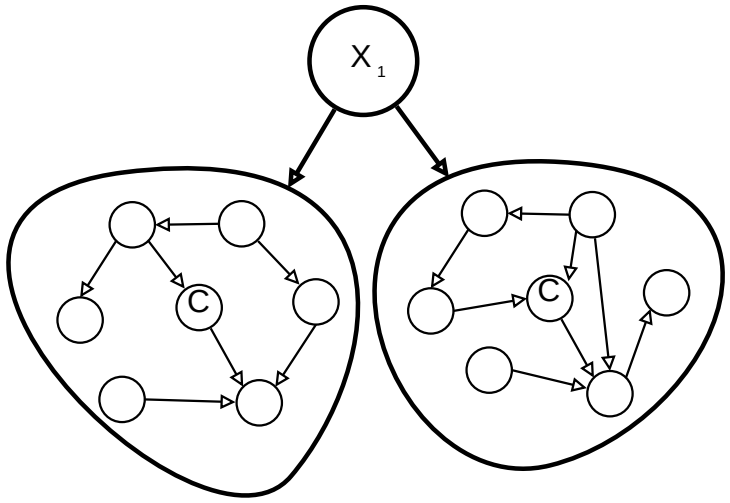


Figura 1.14: Multired bayesiana.

Heckerman en [147] distingue entre dos tipos de independencias asimétricas, las que denomina *de subconjunto* (cuando se trata de una relación entre la variable a clasificar y los atributos) y las *de hipótesis específica* (cuando se trata de una relación entre atributos únicamente). Por tanto, dentro de las

multiredes bayesianas podremos distinguir dos tipos. En aquellas en las que la variable distinguida es la clase, se construirá una red bayesiana para cada uno de los valores de la clase. En el segundo tipo tenemos que la variable distinguida es un atributo (como en el caso de la figura 5.3). Este segundo subtipo pretende modelizar las *independencias asimétricas de hipótesis específica*, mientras que las multiredes cuya variable distinguida es la clase pretenden modelizar las *independencias asimétricas de subconjunto*.

Se denominan multiredes bayesianas recursivas cuando no sólo se escoge un atributo para crear distintas redes sino que se cogen distintos atributos agrupados en forma de árbol teniendo en sus extremos los distintos clasificadores bayesianos, tal y como se puede apreciar en la figura 5.4.

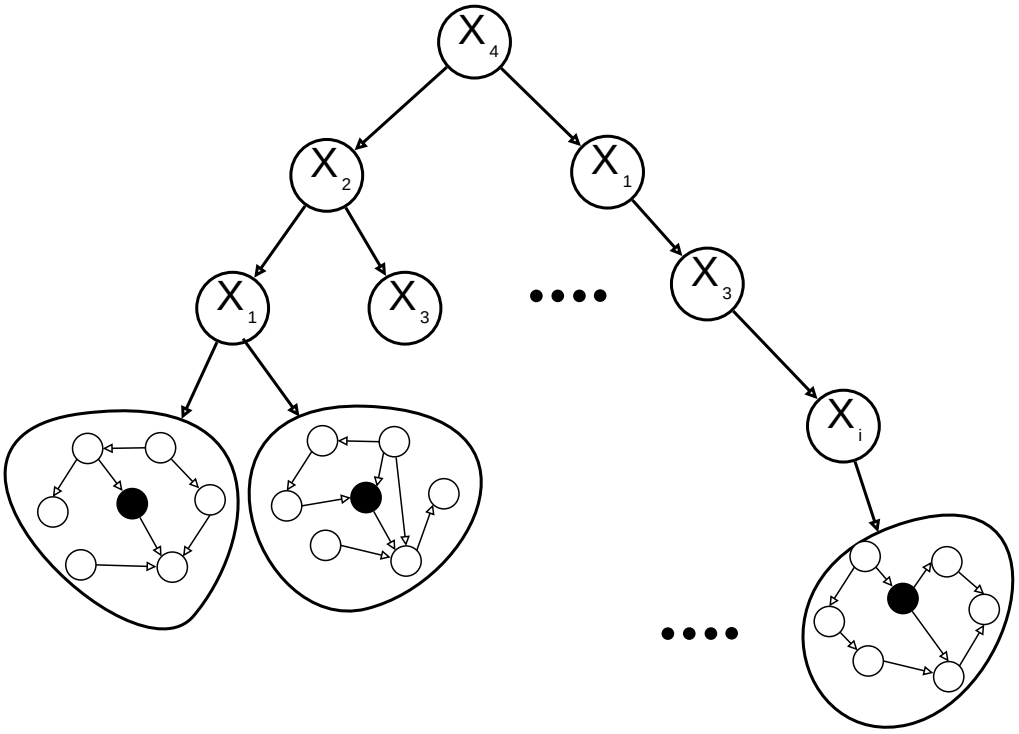


Figura 1.15: Multired bayesiana recursiva.

#### 1.5.4.9. NBTree.

El árbol naïve bayes, *Naïve Bayes Tree (NBTree)*, propuesto por Kohavi en [195], se podría considerar un tipo especial de multired bayesiana recursiva donde las hojas son clasificadores naïve bayes. Aunque la idea que perseguía su autor era combinar las ventajas de los árboles de clasificación [279] y el clasificador naïve bayes.

El procedimiento (ver Algoritmo 8) que sirve para construir el NBTree, empieza con un conjunto de casos $D$ y devuelve un árbol de clasificación con naïve bayes en las hojas.

**Algoritmo 8** Algoritmo para construir el clasificador NBTree

1: Para cada atributo $X_i$ se calcula su utilidad $u(X_i)$. Para los atributos continuos también encuentra un umbral por el cual hacer la división.
2: Sea $j = argmax_i(u_i)$, el atributo con mayor utilidad
3: Si $u_j$ no es lo suficientemente mejor que la utilidad del nodo padre, crea un naïve bayes para el actual nodo del árbol y se sale.
4: Divide el conjunto de casos $D$ de acuerdo a los valores de $X_j$. Si $X_j$ es continua usa el umbral para la división en intervalos.
5: Para cada hijo, se llama al algoritmo recursivamente con la partición correspondiente de $D$

La utilidad de $u(X_i)$ de un atributo $X_i$ se calcula haciendo la suma ponderada de los tantos por ciento de bien clasificados (obtenidos por *validación cruzada de 5-hojas*) de los clasificadores naïve bayes que se construirían si se expandiese ese atributo.

Este algoritmo para de ramificar, además de cuando no hay una mejora significativa al ramificar, cuando la partición del conjunto de casos es menor a un 5 % o cuando el número de casos es menor a 30. No se utiliza ningún método de postpoda después de construir el árbol.

## 1.6. Preprocesamiento de los datos.

Al trabajar con datos del mundo real (no aquellos que son generados de forma sintética) nos encontramos con una serie de problemas al usar redes bayesianas, problemas que deberemos afrontar antes de intentar aprender dichas redes a partir de los datos.

#### 1.6.0.10. Valores perdidos o desconocidos.

La gran mayoría de las técnicas de aprendizaje automático suponen que los datos que se utilizan son completos, es decir, que no presentan valores perdidos o desconocidos, lo cual no siempre es cierto cuando estamos trabajando con datos reales. Por ejemplo, si los datos con los que trabajamos provienen de una encuesta, el encuestado puede negarse a contestar alguna pregunta o puede que desconozca alguna de las respuestas.

La solución a este problema puede consistir en descartar aquellos casos donde se den variables con valores perdidos, que sería la solución más simple pero sólo aplicable en el caso de que tengamos muchos datos completos y cuando la pérdida de información pueda ser irrelevante. Otra solución sería estimar los datos perdidos utilizando algún método de imputación de datos [10]: la idea sería hacer estimaciones de aquellos valores perdidos de forma que no se produzca ninguna pérdida de información útil de los datos. Hay muchas técnicas de imputación de datos perdidos; entre las más comunes, tendríamos crear un nuevo estado para la variable (un estado *desconocido*), usar la media del resto de estados conocidos para esa variable o considerar el valor más probable en base al resto de variables (en clasificación supervisada, en base sólo a la variable a clasificar).

Otra opción para trabajar con valores perdidos es el *algoritmo de maximización de la esperanza* [93] o *algoritmo EM* (por sus siglas en inglés, *expectation-maximization*). El algoritmo EM se usa para calcular los parámetros por estimadores de máxima verosimilitud. Consiste en dos pasos que se repiten de forma iterativa. Paso E (esperanza), donde se estiman los datos perdidos a partir de sus valores esperados. Estos se obtienen utilizando las estimaciones actuales de los parámetros. Paso M (maximización), obtiene estimadores por máxima verosimilitud de los parámetros, considerando los

datos estimados.

El algoritmo EM se utiliza bastante en clasificación no supervisada debido a que se toma la variable clase como una variable de la cual se desconocen todos los valores.

#### 1.6.0.11. Discretización de variables continuas.

Otro de los problemas que se presenta cuando trabajamos con redes bayesianas sobre datos reales, es que en muchas casos aparecen variables discretas (que tienen un número finito de posibles valores, por ejemplo, sexo) y variables continuas (por ejemplo, peso en gramos). Las redes bayesianas normalmente sólo trabajan con variables discretas por lo que necesitamos hacer algo con aquellas variables que son continuas. Esto es debido a que la mayor parte de los algoritmos de propagación o de aprendizaje están perfectamente definidos para variables discretas, pero no para variables continuas. Una solución a este problema pasa por usar variables de tipo continuo usando distribuciones gaussianas [211, 213, 250] o exponenciales de mixturas truncadas [294].

Otra solución pasa por discretizar las variables continuas y tratarlas como si fueran discretas, esto es, transformar los valores de una variable continua en un conjunto de valores discretos [99]. Un algoritmo de discretización divide el dominio de la variable continua en un número finito de intervalos (que serán a la postre los distintos estados discretos) y, posteriormente, a cada observación de la variable continua se le asigna el intervalo que incluye a dicha observación. Por ejemplo, el método más sencillo sería dividir el dominio de la variable continua en intervalos de igual tamaño (tantos intervalos como estados discretos se deseen obtener) y, finalmente, asignar los valores continuos a dichos intervalos. La discretización, si bien es el método más utilizado en la literatura para el tratamiento de variables continuas tiene el problema que si el número de intervalos es demasiado pequeño, se pierde precisión y si es demasiado grande, requiere una gran cantidad de datos para estimar las probabilidades. De esta forma, podemos considerar el problema de encontrar el número de intervalos en un problema de búsqueda.

En el caso de clasificación supervisada la técnica de referencia es el método de discretización por mínima entropía, presentado por Fayyad e Irani [111]. Este método selecciona recursivamente los puntos de corte mediante

un algoritmo de minimización de la entropía entre cada atributo y la clase. Usa el principio de mínima longitud de descripción (MDL) [288].

#### 1.6.0.12. Selección de características.

En la mayoría de los casos el conjunto de variables que tiene un problema suele ser bastante grande y no todas son relevantes. Por ejemplo para determinar si una persona tiene probabilidad de tener un cáncer de pulmón, no es igual de importante una variable que nos diga si es o no fumador, que otra que nos diga el color de ojos. En problemas con un gran número de variables y sobre todo en clasificación supervisada, se aplica una selección de características, también llamada selección de atributos [176, 219, 174]. En este proceso lo que se hace es obtener el mejor subconjunto de variables para trabajar con un problema, eliminando aquellas que no sean útiles o sean redundantes. Las ventajas de este preprocesamiento de los datos son más o menos obvias: con menos variables la red es más fácil de comprender y más rápida de construir, además los clasificadores obtenidos son más exactos -añadir variables irrelevantes o redundantes suele mermar su eficacia-.

Existen dos enfoques para realizar la selección de características: una, mediante *filtrado*(*filter*) donde se seleccionan las variables basándose en medidas sobre los datos y de forma independiente al algoritmo de aprendizaje que se vaya aplicar después; y dos, por *envoltura*(*wrapper*), donde las variables se seleccionan utilizando el algoritmo de aprendizaje como si fuese una caja negra, es decir, durante la selección de características se utiliza el mismo clasificador para evaluar el resultado de escoger unas variables u otras.

## 1.7. Herramienta Elvira.

Muchos de las distintos métodos de trabajo con redes bayesianas mostrados en las secciones anteriores y los que se propondrán a lo largo de este trabajo están implementados en la herramienta *Elvira* [105], disponible en http://leo.ugr.es/elvira/. Elvira es un entorno construido usando el lenguaje de programación Java que permite trabajar con redes bayesianas, diagramas de influencia y muchos otros formalismos (como los árboles de clasificación). Puede operar con variables discretas, continuas y temporales. Además de

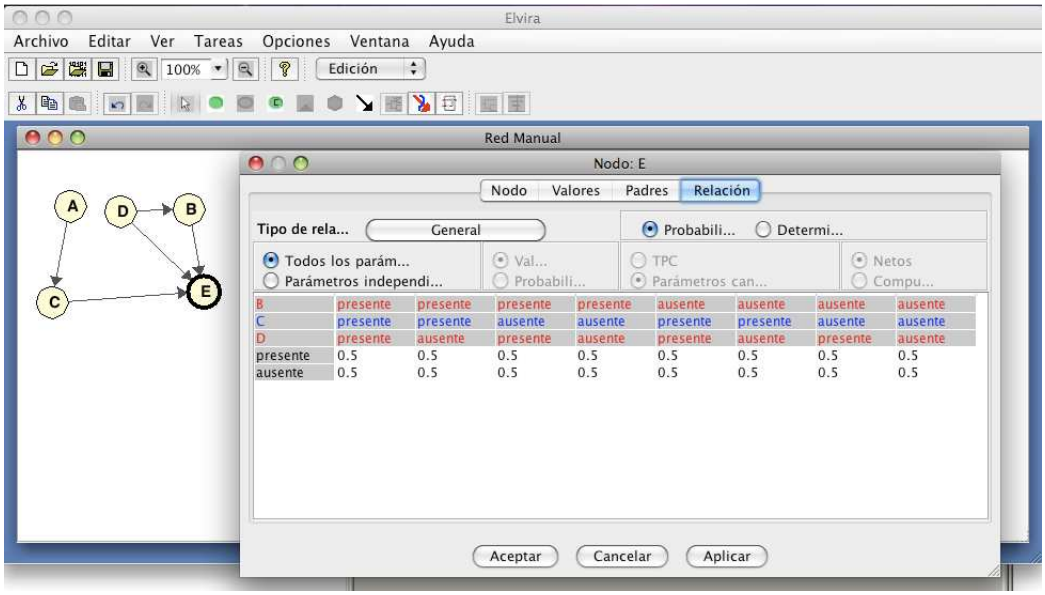


Figura 1.16: Edición manual de una red bayesiana desde la interfaz gráfica de Elvira.

tener un interfaz gráfico de usuario (GUI) fácil de usar, se pueden usar la mayoría de sus algoritmos desde línea de comandos.

El origen del entorno Elvira lo podemos buscar en el interés de distintos grupos de investigación españoles de tener un entorno unificado para el desarrollo de nuevos algoritmos y métodos. De esta forma se evitó tener por duplicado ciertos algoritmos, bien por que realizaban tareas comunes o por que se utilizaban para tomarlos de referencia en resultados experimentales. De esta forma cada grupo de investigación ha ido agregando sus propios algoritmos como resultado de su labor investigadora.

Debido a su origen, Elvira es una herramienta con múltiples usos que nos permite trabajar con conjuntos de datos, grafos, redes bayesianas o diagramas de influencia y un gran conjunto de algoritmos: propagación, clasificación, preprocesamiento, clasificación, abducción, etc.

Desde la interfaz gráfica podemos construir una red bayesiana de forma manual (en la figura 1.16 se muestra la edición manual de las tablas de proba-

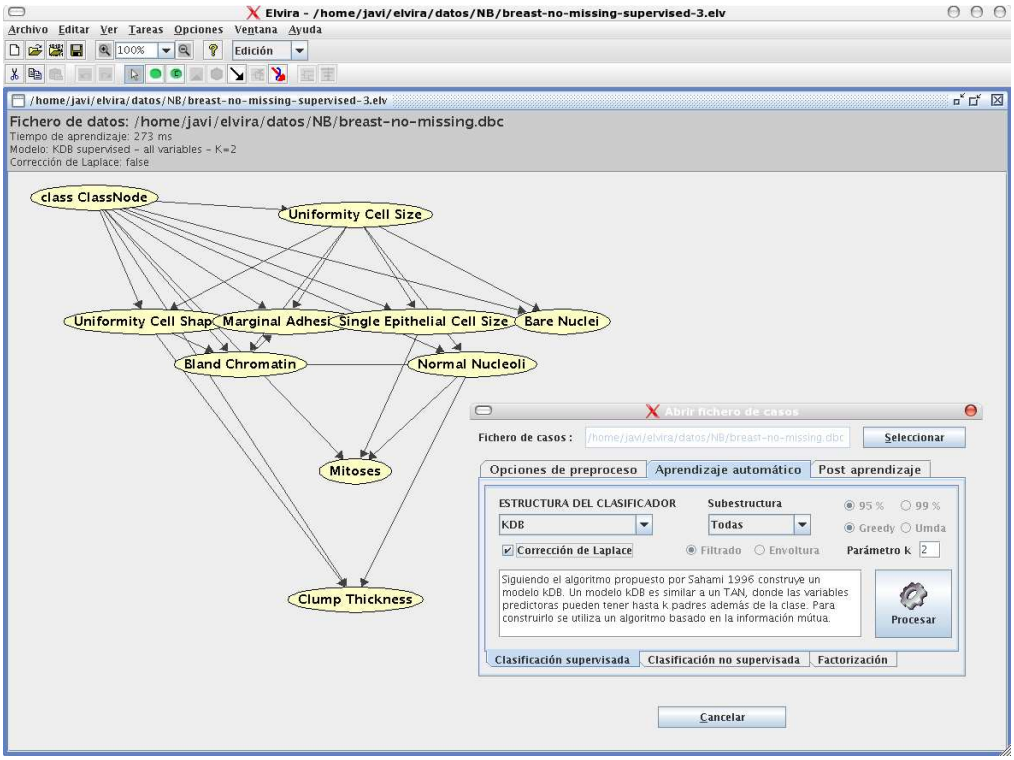


Figura 1.17: Aprendizaje de un clasificador bayesiano k-dependiente a partir de datos.

bilidad condicionadas de un nodo), o usar distintos algoritmos de aprendizaje de redes bayesianas o de clasificadores bayesianos (en la figura 1.17 se puede ver un clasificador bayesiano k-dependiente) o también, por ejemplo, se puede trabajar con variables discretas y continuas (ver figura 1.18).

El interfaz gráfico de Elvira puede operar en tres modos principalmente: edición, inferencia y aprendizaje. El modo de edición es usado para editar o cambiar fácilmente una red o un diagrama de influencia. El modo de aprendizaje nos permite aprender distintos modelos gráficos probabilísticos a partir de conjuntos de datos, incluso nos permite hacer distintos tipos de validaciones para valorar la precisión de los modelos aprendidos. Después de haber realizado la propagación de probabilidades en la red, en el modo de inferencia se puede consultar la probabilidad de cada variable, tanto continua como

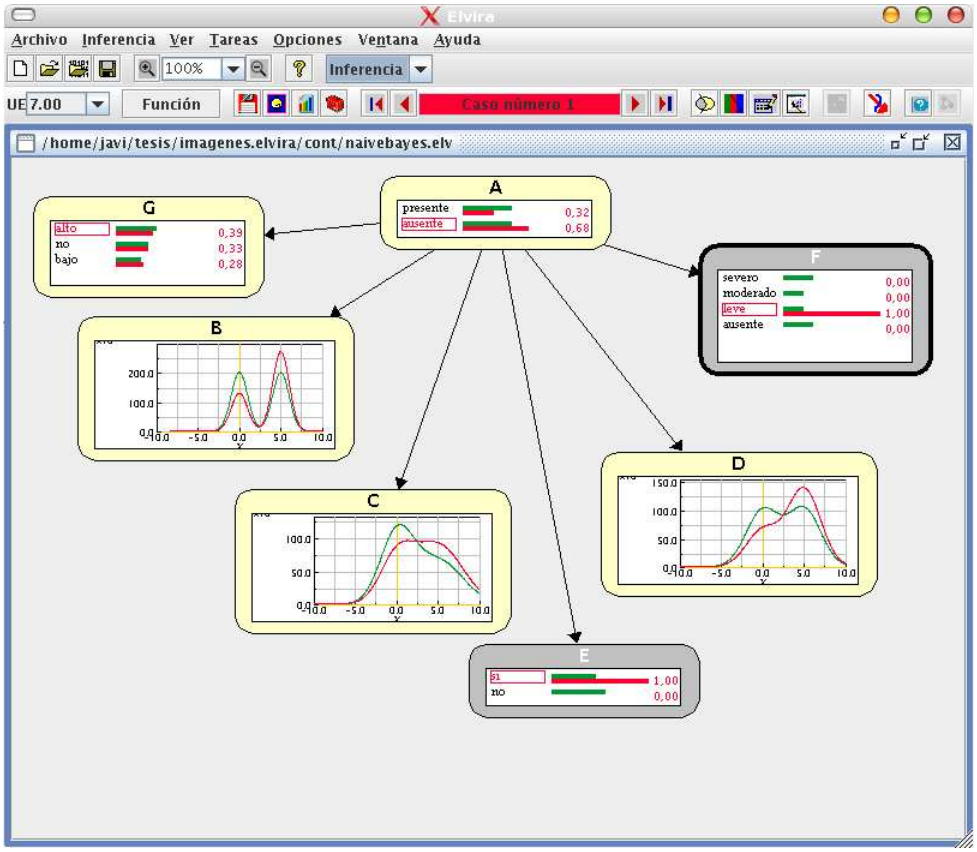


Figura 1.18: Inferencia en una red naïve bayes con variables discretas y continuas.

discreta, y se pueden modificar los valores, si se han observado nuevas evidencias, para obtener la probabilidad a posteriori de un conjunto de variables.

Es necesario destacar el hecho de que la mayor parte de la potencia de Elvira no se encuentra en su interfaz gráfico sino en la biblioteca de clases en Java, debido a que no todos los algoritmos están disponibles desde la interfaz gráfica, aunque sí lo estén desde línea de comandos. Es por esto, que se convierte en un entorno ideal para implementar nuevos métodos apoyándose en el código ya escrito, pudiendo comparar los métodos implementados con otros.

# Capítulo 2

# Datos de expresión genética.

El siglo XXI puede ser considerado como el siglo de la genética, pues podemos ver sus inicios observando los avances del final del siglo XX: la manipulación genética, la clonación genética o la secuenciación de ADN. El principio del siglo XXI comienza con la finalización de la secuenciación del genoma humano, abriendo la puerta al conocimiento de los distintos mecanismos que hacen funcionar nuestro organismo. Aplicándolo en la salud humana, el conocimiento de la información genética nos abre enormes posibilidades al diagnóstico, a la personalización de los medicamentos y al diseño de tratamientos adaptados a las particularidades genéticas de los pacientes.

Dentro de los distintas técnicas utilizadas para poder estudiar el funcionamiento de los genes, tenemos los microarrays de ADN [300] que son una matriz bidimensional de material genético, que nos permiten conocer en un instante determinado -como si fuera una fotografía- qué genes están activados y cuáles no, dentro de una célula. Los datos obtenidos mediante esta técnica se denominan datos de expresión genética. En este capítulo se van a explicar los fundamentos biológicos que hay detrás de la obtención de los datos de expresión genética, qué problemas presentan estos datos y cómo las redes bayesianas se pueden aplicar al anális de la información que contienen estos datos.

Podemos pensar, llegados a este punto, por qué motivo nos hace falta usar redes bayesianas o cualquier tipo de enfoque para estudiar datos de expresión genética. Observemos, simplificando y como apunte inicial, que si consideramos un gen como una variable que toma dos estados (activo y no

activo) y, teniendo en cuenta, que el ser humano tiene entre 20.000 y 25.000 genes, estudiar tantas variables incluso para un problema determinado y para un conjunto reducido de pacientes, puede ser una tarea de enorme complejidad para un experto. Es por ello, que para tratar tal cantidad de datos es conveniente utilizar algún método de extracción de información no trivial y que esté implícita en los datos. Como hemos visto en el capítulo anterior, las redes bayesianas son herramientas con sólidas bases estadísticas y nos van a permitir representar interacciones entre genes. Además su naturaleza probabilística les permite tratar el ruido, muy común cuando estamos trabajando a escala molecular.

## 2.1. La célula.

Antes de seguir hablando de genética es conveniente presentar, al menos brevemente, la célula. Definimos la *célula* como la unidad anatómica, funcional y genética de todo ser vivo, de hecho, la célula es el elemento de menor tamaño que puede considerarse vivo [8]. De este modo, se pueden clasificar los organismos vivos según el número de células que posean: si sólo tienen una, se les denomina *unicelulares* (como, por ejemplo, las bacterias); si poseen más, se les llama *pluricelulares* (como, por ejemplo, el ser humano). En estos últimos el número de células es variable: de unos pocos cientos a cientos de billones, concretamente, $10^{14}$ en el caso del ser humano. La célula posee la capacidad de realizar tres funciones vitales: nutrición, relación y reproducción.

La *teoría celular* [155], propuesta en 1839 por Matthias Jakob Schleiden y Theodor Schwann, postula que todos los organismos están compuestos por células (primer postulado), y que todas las células derivan de otras precedentes (segundo postulado). De este modo, todas las funciones vitales de un organismo ocurren dentro de la célula, en su entorno inmediato o de la interacción entre células adyacentes (tercer postulado). En una célula ocurren todas las funciones vitales, de manera que basta una sola de ellas para tener un ser vivo (que será un ser vivo unicelular). Así pues, la célula es la unidad fisiológica de la vida. Además, la tenencia de la información genética en el *ácido desoxirribonucleico* (*ADN*) permite la transmisión de esa información a la siguiente generación celular (cuarto postulado).

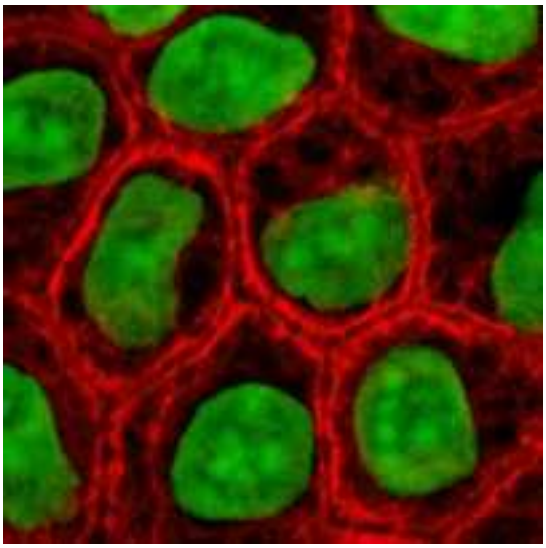

Figura 2.1: Células de la piel humana coloreadas, de verde el núcleo y de rojo la membrana plasmática (Fuente Wikimedia http://commons.wikimedia.org/wiki/File:Epithelial-cells.jpg).

Existen dos grandes tipos celulares: las *procariotas* (por ejemplo, las bacterias) y las *eucariotas* (por ejemplo, las que poseen animales y vegetales). Las eucariotas son aquellas células que tienen la información genética envuelta dentro de una membrana que forman el llamado *núcleo*. Por otro lado, muchos seres unicelulares tienen la información genética dispersa por el interior de la célula, no tienen núcleo. A ese tipo de células se les da el nombre de procariotas. Las células procariotas suelen ser más pequeñas y menos complejas que las eucariotas, como se puede ver en la figura 2.2.

La estructura común a todas las células comprende la membrana plasmática, el citoplasma y el material genético o ADN.

- La *membrana plasmática* o *citoplasmática* es una estructura laminar que engloba a las células, define sus límites y contribuye a mantener el equilibrio entre el interior y el exterior de éstas como se ve en la figura 2.1.
- El *citoplasma*: abarca el medio líquido, o *citosol*, y el *morfoplasma* (nombre que recibe una serie de estructuras denominadas *orgánulos*

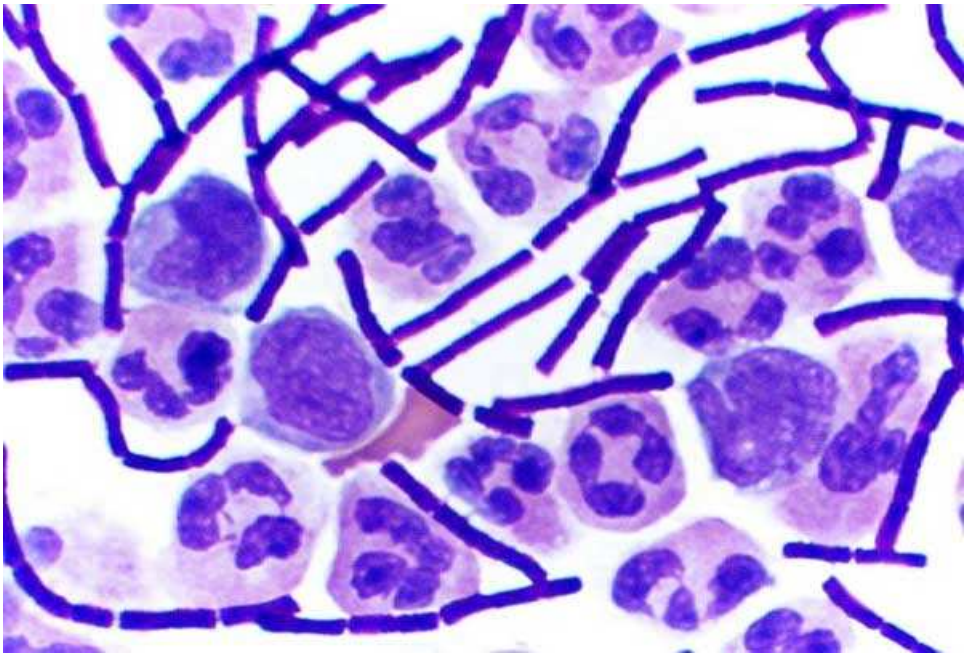

Figura 2.2: Comparativa de tamaño entre células sanguíneas eucariotas, que son de mayor tamaño, y bacterias, procariotas, que son de menor tamaño, con forma de bastón (Fuente Wikimedia http://commons.wikimedia.org/wiki/File:Gram_Stain_Anthrax.jpg).

celulares). El citosol es la sede de muchos de los procesos metabólicos que se dan en las células.

- El material genético: donde tenemos los cromosomas que son segmentos largos de ADN donde se almacenan los genes. El material genético puede estar o no rodeado por una membrana formando el núcleo (sólo en las células procariotas), tal y como se aprecia en la figura 2.3.

Las células eucariotas, además de la estructura básica de la célula (membrana, citoplasma y material genético) presentan una serie de estructuras fundamentales para sus funciones vitales:

- El *sistema endomembranoso*: el sistema endomembranoso es el sistema de membranas internas que divide la célula en compartimientos funcionales y estructurales, denominados *orgánulos*.

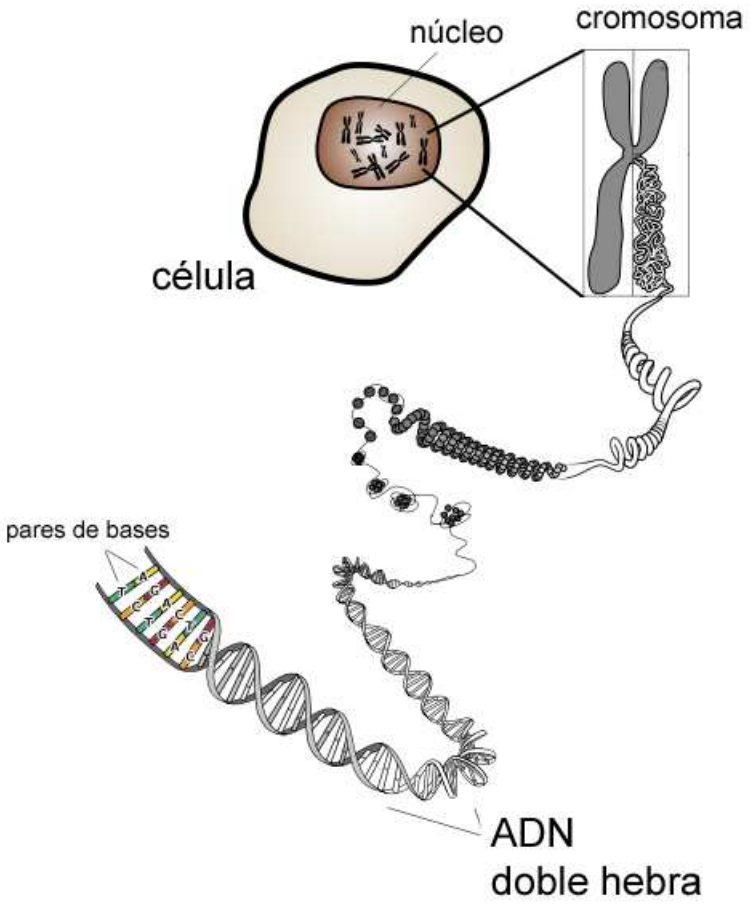


Figura 2.3: Situación de los cromosomas y el ADN en células eucariotas (Fuente Wikimedia http://commons.wikimedia.org/wiki/File:Chromosome_Spanish_text.png).

- Orgánulos transductores de energía: son las *mitocondrias* y los *cloroplastos*. Su función es la producción de energía a partir de la oxidación de la materia orgánica (mitocondrias) o de energía luminosa (cloroplastos). Los cloroplastos sólo están presentes en organismos fotosintetizadores como, por ejemplo, las plantas.

- Estructuras carentes de membranas: están también en el citoplasma y son los *ribosomas*, cuya función es sintetizar proteínas; y el *citoesque-*

*leto*, que es una red de filamentos proteicos, que le confieren forma y organización interna a la célula y permiten su movimiento, además de permitir el movimiento de las moléculas y orgánulos en el citoplasma.

- El *núcleo*: mantiene protegido al material genético.

En el exterior de la membrana plasmática de la célula procariota se encuentra la *pared celular*, que protege a la célula de los cambios externos. El interior celular es mucho más sencillo que en las eucariotas; en el citoplasma se encuentran los ribosomas, prácticamente con la misma función y estructura que las eucariotas. También se encuentran los *mesosomas*, que son bolsas formadas por pliegues de la membrana. No hay, por tanto, citoesqueleto ni sistema endomembranoso.

## 2.2. Los genes.

La información de los caracteres hereditarios se encuentran en los *genes* [139]. Un gen se define como una secuencia de *ácido desoxirribonucleico* (*ADN*), que es usada para construir los *aminoácidos* que forman una *proteína*. Las proteínas, a su vez, ocupan un lugar de máxima importancia entre las moléculas constituyentes de los seres vivos. Prácticamente todos los procesos biológicos dependen de la presencia o la actividad de las proteínas. Por un lado, un gen puede generar una *proteína estructural* que contribuye a las propiedades físicas de la célula o el organismo, por ejemplo las proteínas que forman los músculos. Por otro lado, un gen puede generar una *enzima* que cataliza una de las reacciones de la célula. Por tanto, al estar las proteínas codificadas en los genes, determinan los dos aspectos importantes de la estructura y función biológicas. No obstante, los genes no dictan ellos solos la estructura de un ser vivo (ya que un elemento crucial es el medio ambiente), aunque sí lo condicionan.

Dentro de un gen que codifique una proteína, los *exones* son aquellas partes del gen que contienen la información para producir la proteína. Cada exón codifica una parte específica de la proteína completa, de manera que el conjunto de exones forma la región codificante del gen, como se puede ver en la figura 2.4. En los organismos eucariotas los exones de un gen están separados por regiones largas de ADN, llamadas *intrones*, que tienen otras funciones distintas a codificar proteínas.

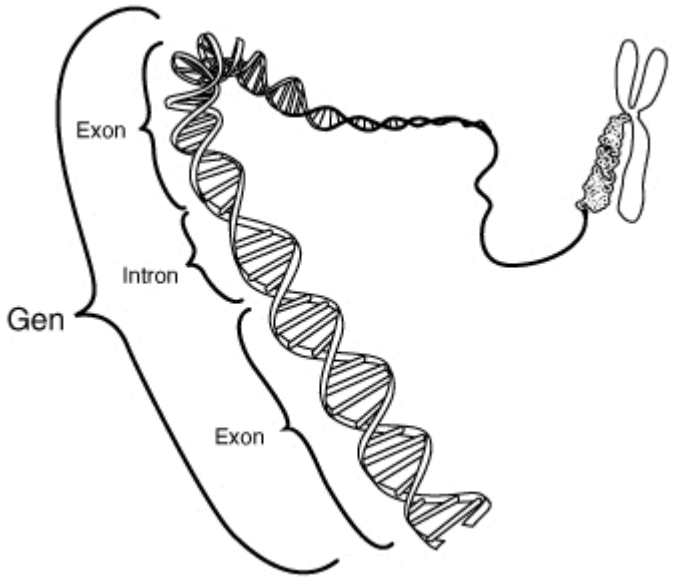


Figura 2.4: Partes de un gen (Fuente Wikimedia http://commons.wikimedia.org/wiki/File:Gene.png).

Los genes se disponen a lo largo de cada uno de los *cromosomas* de la célula. Cada gen ocupa en el cromosoma una posición determinada llamada *locus*. Los organismos *diploides* (entre ellos, casi todos los animales y plantas) disponen de dos juegos de cromosomas, cada uno de ellos proveniente de uno de los padres. Así pues, cada par de cromosomas tiene dos copias de cada gen, una procedente de la madre y otra del padre. Los genes pueden aparecer en versiones diferentes, con variaciones pequeñas en su secuencia, y entonces se los denomina *alelos*. Los alelos pueden ser dominantes o recesivos. Cuando una sola copia del alelo hace que se manifieste el rasgo genético heredado, el alelo es dominante. Cuando son precisas dos copias del alelo (una en cada cromosoma del par), el alelo es recesivo. El ser humano contiene un total de 23 pares de cromosomas (como se puede observar en la figura 2.5), de los cuales dos de ellos determinan el sexo.

El conjunto de cromosomas de una especie se denomina *genoma*, aunque en realidad el genoma es todo el material genético contenido en las células de un organismo en particular. Por lo general, al hablar de genoma en los seres

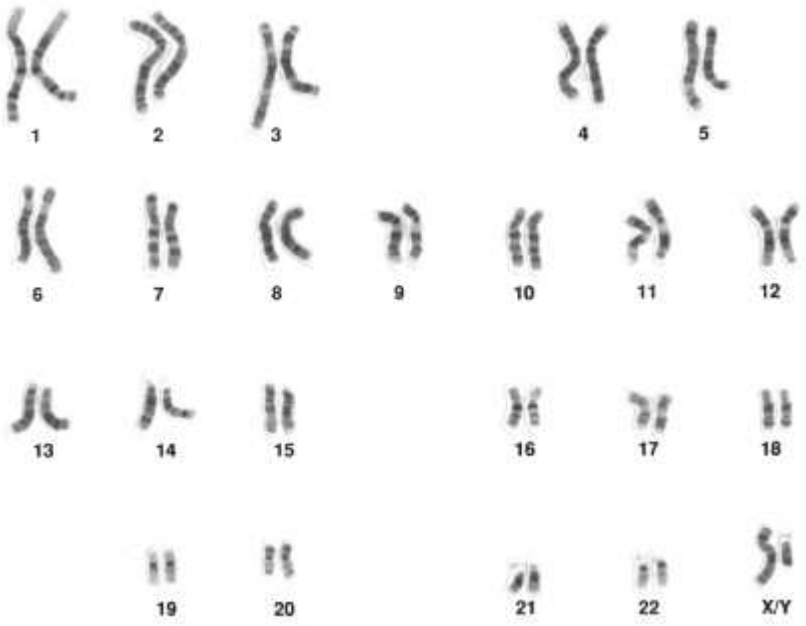


Figura 2.5: Cromosomas de una persona de sexo masculino, con 46 cromosomas incluidos los cromosomas sexuales XY (Fuente Wikimedia http://commons.wikimedia.org/wiki/File:Human_male_karyotpe.gif).

eucarióticos nos referimos sólo al ADN contenido en el núcleo, organizado en cromosomas pero no debemos olvidar que también la mitocondria (otro orgánulo de la célula) contiene genes.

El proyecto del genoma humano [1] fue un proyecto de investigación internacional con el objetivo de determinar la secuencia completa de ADN del ser humano e identificar, de esta forma, los aproximadamente 20.000-25.000 genes del genoma humano. El proyecto empezó en 1990 y en 2003 mostró una primera versión del genoma humano, no obstante, hoy en día se sigue investigando en este proyecto, incluso se ha extendido a identificar los genomas de otras especies.

Tanto interés por el genoma humano es debido a los beneficios que ello nos puede traer, por ejemplo:

[1] `http://www.ornl.gov/sci/techresources/Human_Genome/home.shtml`

- Diagnóstico y prevención de enfermedades: las pruebas basadas en el ADN -prueba genética- se pueden usar para el diagnóstico de enfermedades; pronóstico y evolución de la enfermedad; confirmar la presencia de enfermedad en pacientes asintomáticos; y, con un grado de incertidumbre, para predecir el riesgo de enfermedades en personas sanas y en su descendencia. Por ejemplo, podemos obtener una probabilidad de padecer algún tipo de cáncer.

- Intervención o tratamiento sobre la enfermedad: conociendo la causa de la enfermedad podemos desarrollar técnicas para tratar enfermedades hereditarias.

## 2.3. Ácido desoxirribonucleico (ADN).

El *ácido desoxirribonucleico*, abreviado como *ADN* (y también *DNA*, del inglés *DeoxyriboNucleic Acid*), como hemos mencionado anteriormente, contiene la información genética usada en el desarrollo y el funcionamiento de las células (y también en los virus[2]), siendo el responsable de su transmisión hereditaria.

Desde el punto de vista químico, el ADN está formado por la unión de pequeñas subunidades denominadas *nucleótidos*; es un polímero de nucleótidos, es decir, un polinucleótido. Un *polímero* es un compuesto formado por muchas unidades simples conectadas entre sí, como las cuentas de un collar. Un nucleótido está formado por un azúcar, (la *desoxirribosa*), una base nitrogenada y un grupo fosfato que actúa como enganche de cada parte con la siguiente. Las cuatro bases nitrogenadas que se encuentran en el ADN son la *adenina* (abreviado A), *citosina* (C), *guanina* (G) y *timina* (T). Cada una de estas cuatro bases está unida al armazón de azúcar-fosfato a través del azúcar para formar el nucleótido completo (base-azúcar-fosfato). Estructuralmente la molécula de ADN se presenta en forma de dos cadenas helicoidales de nucleótidos enrolladas alrededor de un mismo eje (imaginario); las cadenas están unidas entre sí en pares de bases. Los emparejamientos son siempre adenina-timina y citosina-guanina, tal y como se puede apreciar en la figura

[2]Un *virus* es una entidad biológica que necesita de una célula para poder reproducirse. Poseen una carga genética en forma de ADN o ARN. Los biólogos debaten si son organismos vivos o no, pero los podemos situar en el límite entre la materia viva y la materia inerte.

2.6.

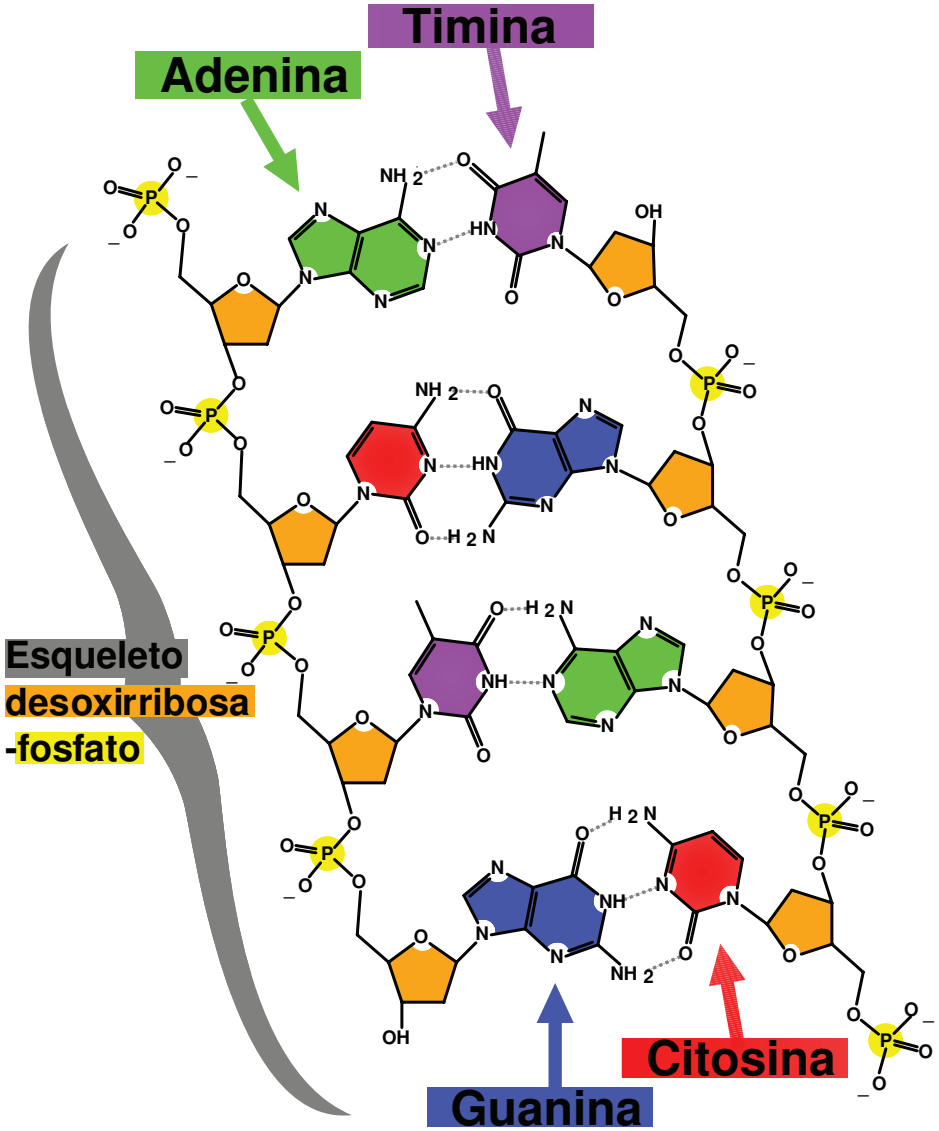


Figura 2.6: Estructura del ADN (Fuente Wikimedia http://commons.wikimedia.org/wiki/File:DNA_chemical_structure_es.svg).

Lo que distingue a un nucleótido de otro es la base nitrogenada, y por ello

la secuencia del ADN se especifica nombrando sólo la secuencia de sus bases. La disposición secuencial de estas cuatro bases a lo largo de la cadena es la que codifica la información genética: por ejemplo, una secuencia de ADN puede ser CGATGCCTCGAA...

Para que la información que contiene el ADN pueda ser utilizada, debe copiarse en primer lugar en otros polímeros de nucleótidos, más cortos y con unas unidades diferentes, llamados *ARN* (*ácido ribonucleico* o, en inglés *RiboNucleic Acid -RNA-*). El ARN es monocatenario y en lugar de tener la base nitrogenada timina (T), tiene la base *uracilo* (U). Las moléculas de ARN se copian exactamente del ADN mediante un proceso denominado *transcripción*. Una vez realizada la transcripción (en el núcleo, si estamos en una célula eucariota), las moléculas de ARN pasan al citoplasma para su utilización posterior, por eso a esta cadena se le llama ARN mensajero o, abreviadamente, ARNm. El ARNm madura, lo cual significa que se produce la pérdida de las secuencias no codificantes, esto es, los intrones. La información contenida en el ARN se interpreta usando el *código genético*. El código genético especifica la secuencia de los aminoácidos de las proteínas, según una correspondencia de un triplete de nucleótidos (lo que se denomina *codón*) para cada aminoácido, tal y como se puede ver en la figura 2.7, es decir, el código genético asigna a cada codón un aminoácido.

En resumen, la información genética se halla codificada en las secuencias de nucleótidos del ADN y debe traducirse para poder ser empleada. Tal traducción se realiza empleando el código genético a modo de diccionario. Por ejemplo, siguiendo el ejemplo de la figura 2.7 y el ejemplo anterior de la cadena de ADN CGATGCCTCGAA..., ésta se transcribiría en una molécula de ARN que se leería GCU-ACG-GAG-CUU-... empleando como molde la cadena complementaria del ADN antes citada; el ARN resultante, utilizando el código genético, se traduciría como la secuencia de aminoácidos alanina-treonina-ácido glutámico-leucina-...

## 2.4. Microarrays de ADN.

Los microarrays de ADN [300, 221, 19] (también llamados chips de ADN) están formados, generalmente, por un substrato de cristal o de nylon, donde se ordenan densamente, en forma de matriz, cadenas simples de oligonucleótidos de ADN (a estas muestras inmovilizadas en el soporte se les denomina *sondas*). Un *oligonucleótido* es una secuencia corta de ADN o ARN, con

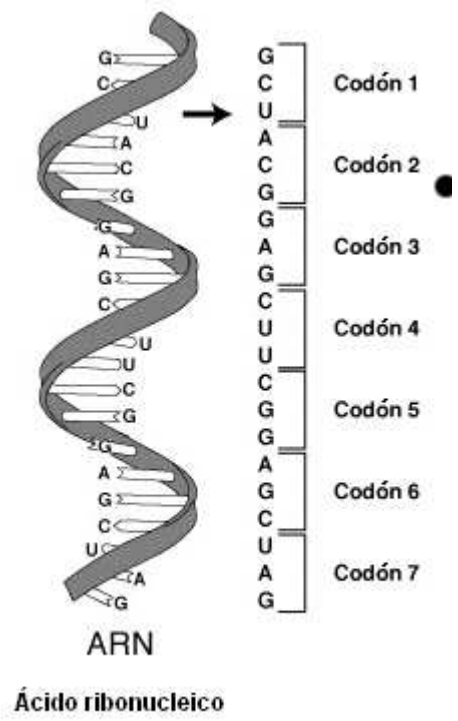


Figura 2.7: Representación de codones en una cadena de ARN (Fuente Wikimedia http://commons.wikimedia.org/wiki/File:Codones-ARN.png).

cincuenta o menos pares de bases, es decir, son pequeños trozos de una secuencia específica de ADN, puede ser un pequeño trozo de un gen u otra parte del ADN usado como muestra para que se hibriden con otras muestras de ADN o ARN complementario, abreviadamente, ADNc y ARNc. El ADNc y el ARNc se caracterizan porque no poseen intrones, lo que implica que son más simples y fáciles de hibridar.

Si se pone el microarray en contacto con una muestra que posea una mezcla de cadenas simples de ADN a identificar, éstas se hibridarán con su oligo complementario, de esta forma en el microarray se quedarán las hibridaciones de aquellos genes que estén activos en las células. El nivel de muestras hibridadas en cada sonda del microarray se mide mediante un escáner (normalmente con una luz fluorescente) que determina la abundancia o escasez

de dichas muestras. De esta forma se pueden observar genes que se activan o se reprimen en distintas condiciones o distintas muestras. La contrapartida de estos experimentos es que no se pueden observar niveles absolutos en la expresión. En la figura 2.8 cada punto representa una sonda y los distintos tonos de colores indican la expresión del gen. A la postre, los datos de expresión genética serán valores numéricos continuos representando los distintos colores y su intensidad.

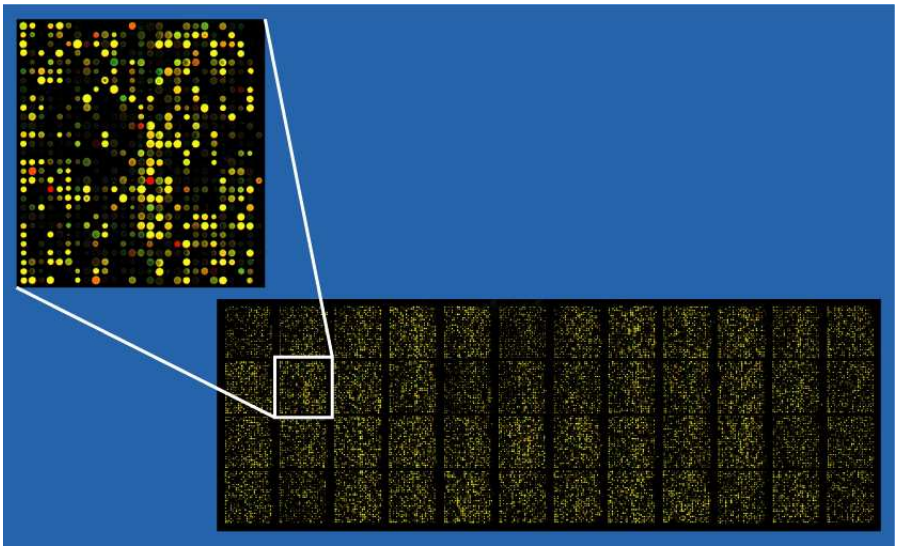

Figura 2.8: Ejemplo de un microarray de ADN con 40000 sondas (Fuente Wikimedia http://commons.wikimedia.org/wiki/File:Microarray2.gif).

En un microarray de ADN pueden ser analizados desde diez a miles de genes. Al poder estudiar tantos genes a la vez, esta técnica se ha aplicado para un mayor entendimiento de varios procesos biológicos como son la identificación de los genes implicados en complejos procesos humanos, como, por ejemplo, el cáncer [9, 36, 191] y la búsqueda de nuevos fármacos [292] o en el estudio de diferentes procesos en otros organismos [319, 94]. Téngase en cuenta que la disponibilidad de datos de expresión genética nos va a permitir comprender procesos celulares desconocidos, realizar diagnósticos y tratamientos de enfermedades y diseñar fármacos con una función conocida a nivel genético.

No se puede establecer un estándar en microarrays de ADN, ya que hay


Figura 2.9: Microarrays de la marca Affymetrix para el genoma humano (izquierda) y para el genoma del ratón(derecha) (Fuente Wikimedia http://commons.wikimedia.org/wiki/File:Affymetrix-microarray.jpg).

distintas tecnologías y fabricantes con el mismo fin. Affymetrix (ver figura 2.9) es el líder del mercado, sobre todo, en la venta de microarrays de alta densidad (capaz de mirar la expresión de gran cantidad de genes a la vez). Sin embargo, y para la práctica clínica, donde se necesitan microarrays de menor densidad y coste, existen otras compañías como, por ejemplo, Agilent (ver figura 2.10).

Debido a las posibilidades que los microarrays nos brindan, en la literatura existente hay una gran cantidad de métodos propuestos para su análisis: análisis de componentes principales [285], *Support Vector Machines* (SVM) [55], métodos de agrupamiento (*clustering*) [103], agrupamiento jerárquico [12], mapas auto-organizativos [136], etc. Y también redes bayesianas[122,

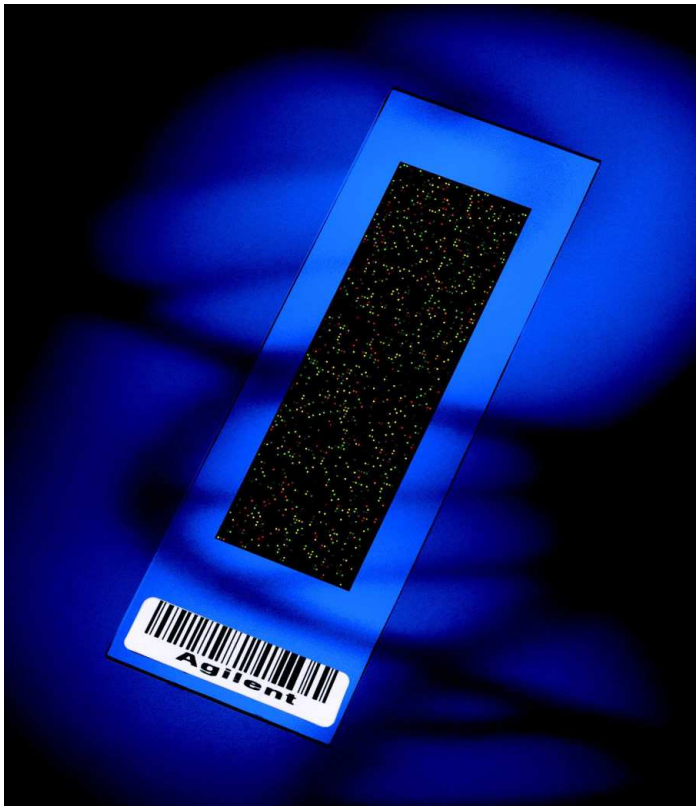


Figura 2.10: Microarray de la marca Agilent -los colores son una representación de cómo se vería en el escáner, en realidad es transparente-.

116] . Los trabajos en este sentido los veremos en el apartado 2.5.

### 2.4.1. Aplicaciones de los microarrays de ADN.

Desde que el *proyecto del genoma humano* presentara la secuencia completa de ADN del ser humano, la demanda de tecnologías que faciliten la búsqueda de genes y sus patrones de expresión ha sido espectacular. Una vez identificados los genes, el siguiente paso es estudiar su comportamiento. Como hemos visto, los microarrays de ADN nos permiten el análisis simultáneo de miles de genes, por lo que es una herramienta adecuada para estudiar el comportamiento de los genes pero también para muchas más aplicaciones.

**Análisis de la expresión genética.**

Los datos de expresión genética son útiles para conocer las bases moleculares de un organismo, por ejemplo, en el estudio de la célula. Si bien se puede ampliar su utilización para llegar a la comprensión de tejidos y órganos o enfermedades de base molecular como el cáncer. También se pueden utilizar para el diagnóstico y tratamiento de enfermedades con patrones identificables de expresión genética -denominados *marcadores*- , permitiendo así la identificación y pronóstico de dicha enfermedad. Sin embargo, los datos procedentes del análisis de la expresión genética también tienen sus inconvenientes, ya que, por ejemplo, la expresión de un gen no siempre tiene una consecuencia fisiológica.

Veamos distintas áreas de interés donde se puede aplicar el análisis de datos de expresión genética [221]:

- *Células y tejidos normales*: la expresión de un gen proporciona información indirecta acerca de su función, por lo que si descubrimos los procesos que desencadenan la expresión de dicho gen, podremos diseñar fármacos que lo activen o desactiven, minimizando los efectos secundarios.

- *Patologías*: la regulación negativa o positiva de la expresión de un gen puede ser la causa de la patología o bien el resultado de la misma. Si se estudia la función de aquellos genes activados, inhibidos o mutados cuando se comparan tejido sano y enfermo, podremos encontrar la causa de la enfermedad e identificar potenciales dianas terapéuticas.

- *Modelos*: el uso de animales -por ejemplo, ratones transgénicos- como modelos proporciona gran cantidad de información y la posibilidad de comparar sus patrones genéticos de expresión.

- *Patógenos*: una de las ventajas que presenta el trabajo con genomas procedentes de microorganismos es su pequeño tamaño, lo que ha permitido la secuenciación de un gran número de ellos en poco tiempo. Por tanto, se puede realizar el estudio de la expresión de los genes del patógeno, para estudiar el curso de una infección o la respuesta del organismo frente al patógeno.

**Estudio de nuevos fármacos y validación de dianas terapéuticas.**

Como hemos visto, los datos provenientes de microarrays nos permiten encontrar potenciales dianas terapéuticas en las enfermedades, conocer mejor el mecanismo de acción de los elementos, mejorar su eficacia o determinar posibles efectos secundarios.

En este campo se persigue en primer lugar, la identificación de dianas terapéuticas [92, 258] que nos permitan avanzar en el tratamiento de patologías.

En segundo lugar, se busca la implementación de tecnologías que aumenten la eficiencia del proceso de descubrimiento de patrones de expresión de la actividad celular y desarrollo de nuevos fármacos [79, 343].

**Medicina personalizada (farmacogenómica).**

La presencia de genes alternativos o expresiones atípicas de genes implicados en la acción de un fármaco puede provocar una resistencia a la terapia o bien una respuesta atípica a la misma. La *farmacogenómica* estudia cómo la genética de un paciente determina la respuesta frente a una terapia determinada. Por tanto, la información obtenida con estudios farmacogenómicos se utiliza para diseñar microarrays de ADN que puedan ser utilizados en la selección de fármacos a medida, y así obtener un fármaco más efectivo que minimice los efectos secundarios.

**Pronóstico y diagnóstico de enfermedades.**

La obtención de información mediante el uso de microarrays de ADN puede sernos muy útil en el pronóstico y diagnóstico de enfermedades.

La detección de mutaciones y polimorfismos permite el estudio de las variantes de un mismo gen y la detección de mutaciones en genes que participan en enfermedades complejas. En este tipo de microarrays, lo que se fija al soporte son oligonucleótidos sintéticos con pocas bases, donde cada uno representa un fragmento de un determinado gen, de modo que se puedan localizar variantes y mutaciones de genes conocidos.

También existen microarrays de diagnóstico y caracterización tumoral, donde se utilizan secuencias que contienen distintas mutaciones en protooncogenes (genes cuyos productos promueven el crecimiento y la división celular) o en genes supresores de tumor.

**Detección de agentes infecciosos.**

Por otro lado, se encuentran los microarrays para diagnóstico molecular de enfermedades infecciosas como, por ejemplo, el SIDA o la hepatitis C que permiten el genotipado de las cepas infecciosas a partir de muestras del paciente.

### 2.4.2. Problemas y características de los datos de expresión genética.

Una vez que las hebras de ADN de la muestras se han hibridado, pasamos el microarray por el escáner para determinar el grado de expresión de cada gen, y obtendremos finalmente un valor numérico con los niveles de expresión para cada gen estudiado. Esto que, así dicho, pueda parecer un proceso fácil de realizar, conlleva muchos problemas desde el punto de vista biológico (por ejemplo, diseño del microarray o normalización entre las distintas pruebas).

Debido a que los problemas en la obtención de los datos de expresión genética se escapan del ámbito de este trabajo, nos vamos a centrar en los problemas de los datos ya obtenidos, los cuales están compuestos por un conjunto de números reales (nivel de expresión de cada gen) para cada individuo estudiado. Encontramos distintos tipos de problemas en el análisis de estos datos de expresión genética [322]:

- *Número de casos versus variables*: los microarrays de ADN son una tecnología cara de usar, lo que provoca que en los estudios, el número de muestras -casos- que se tienen sea bastante reducido. Por otro lado, los datos que se generan con microarrays tienen una gran cantidad de variables (como se puede ver en la figura 2.8), una por cada gen que se está estudiando. Por lo tanto tenemos un pequeño número de casos con una gran cantidad de variables a analizar, son un claro ejemplo de problemas *n grande, m pequeña* [355], donde $m$ es el número de muestras y $n$ el número de variables.

- *Errores en la muestra y ruido*: al trabajar a escala molecular, y debido a las imperfecciones de la tecnología actual, cualquier pequeña variación, error o agente externo, como una mota de polvo, puede provocar que los datos aportados por el microarray no sean del todo correctos o que tenga cierta cantidad de ruido [190].

- *Valores perdidos*: se suelen producir por diversos motivos, incluyendo resolución insuficiente, corrupción de la imagen, o simplemente polvo o un pequeño rasguño en el cristal. También ocurren de forma sistemática por los métodos utilizados para crear los microarrays [341].

- *Ciclo celular*: debido a que cada gen se activa en un momento dado dentro del ciclo celular, no es de extrañar pensar que células de una misma muestra se encuentren en distintas fases de su metabolismo teniendo, por tanto, diferentes expresiones genéticas [319].

- *Promediación* (*Averaging*): la muestra con la cual se hibridan las sondas del microarray está compuesta por distintas células, por tanto, el resultado en una sonda del microarray no es el valor de una célula en particular sino un valor medio del conjunto que forma la muestra a analizar.

- *Datos continuos*: los valores numéricos que miden la expresión de cada gen son continuos, por tanto, puede ser deseable discretizarlos, como paso previo a la aplicación de una metodología que se vaya a utilizar para análisis.

- *Normalización*: en los experimentos de microarrays hay muchas fuentes de variación sistemática (como por ejemplo, diferencias en las cantidades puestas en las sondas, hibridación dispareja o precisión del escáner). Se denomina normalización [365, 283] al proceso de eliminación de tales variaciones. Normalmente, se suele normalizar durante la adquisición de los datos pero hay que tener en cuenta que dos conjuntos de datos de distintos experimentos aunque se estudien los mismos genes bajo las mismas circunstancias pueden no ser compatibles entre sí.

## 2.5. Datos de expresión genética y redes bayesianas.

Una *red de regulación genética*, o simplemente *red genética*, es una representación en forma de red de las interacciones entre genes, cuyo conocimiento es fundamental para estudiar el comportamiento de la célula. Los nodos de esta red son los genes, y las conexiones entre ellos representan los intrincados

mecanismos de regulación de la expresión genética, de tal forma que un enlace entre dos genes puede indicar que los cambios en un gen provocan cambios en el otro, esto es, uno regula positiva o negativamente la expresión del otro.

Obtener una red genética a partir de una red bayesiana es un proceso inmediato: en la red bayesiana las variables representan los genes de la red genética y los enlaces representan relaciones de regulación (activaciones o inhibiciones) como en las redes genéticas. Aunque también hay algunas diferencias: una, en una red bayesiana no pueden aparecer ciclos como sí aparecen en las redes de regulación genética; dos, un enlace puede existir en la red bayesiana pero no en la red genética, en ese caso, dicho enlace puede indicar que ambos genes se expresan a la vez, es decir, están corregulados.

Las redes bayesianas se pueden obtener, como sabemos, de forma manual con la ayuda de un experto o de forma automática con un algoritmo de aprendizaje. En este segundo caso se utilizan datos de expresión genética (normalmente obtenidos a partir de microarrays de ADN).

Obtener redes de regulación genética a partir de datos de expresión genética es un área de gran interés en bioinformática y hay distintas aproximaciones para obtener redes genéticas a partir de estos datos [327, 226]. No obstante, las redes bayesianas han sido la técnica que mejores resultados ha obtenido [358, 295, 23] aunque, si bien, se les achaca un mayor coste computacional [351].

La expresión de los genes es un proceso temporal que varía a lo largo del ciclo celular, es por ello que se ha considerado el uso de redes bayesianas dinámicas para modelizar redes genéticas teniendo en cuenta esa variación temporal. Además las redes bayesianas dinámicas a diferencia de las estáticas permiten modelizar ciclos que son muy comunes en los procesos biológicos.

No sólo es interesante el uso de redes bayesianas a partir de datos de microarrays de ADN para conocer con más profundidad el comportamiento de la célula, también pueden ser muy útiles en clasificación, por ejemplo, para agrupar genes con funcionalidades comunes, predecir distintos tipos de patologías o para centrar nuestro estudio en el conjunto de genes más relevantes respecto a una función biológica determinada. Para estos casos se usan clasificadores bayesianos.

Pero no sólo se han utilizado las redes bayesianas para la obtención de re-

des genéticas o en la clasificación de los datos de expresión genética, también se han utilizado en el análisis del procedimiento en la obtención de microarrays de ADN: en el nivel de preparación de las sondas, Tobler y col. en [340] utilizan un clasificador naïve bayes entre otros para la predicción de sondas de ADN usados en los microarrays. Debido a que de las sondas depende la calidad de la hibridación en los microarrays, encontrar buenas sondas es una tarea difícil. Hay que tener en cuenta que el tamaño de las sondas de ADN utilizadas en la hibridación de los genes es sólo una pequeña parte de la longitud del gen. Usando buenas sondas (que hibriden bien) tenemos que poner menos por gen en los microarrays y así podremos medir más genes.

En el nivel de análisis de imágenes, las redes bayesianas sobre datos de píxeles se han usado para representar y mejorar la calidad de las medidas [146].

En el nivel del procesamiento de las medidas, Baldi y col. en [20] muestran como un test-t bayesiano puede ser usado para estimar las distribuciones de las medidas de expresión genética, viendo como este enfoque puede compensar la ausencia de más experimentos; luego en [220] se aplica satisfactoriamente a experimentos sobre la bacteria *Escherilia Coli*. En [165] se utiliza una aproximación parecida pero con la diferencia que la estructura entre genes es modificada. En cambio, en [54] se usa un modelo jerárquico para tener en cuenta múltiples niveles discretos en los que la expresión de los genes puede cambiar. Por otro lado, English y col. [106] utilizan las redes bayesianas para identificar las fuentes de ruido en experimentos de microarrays.

En [116] encontramos un buen trabajo introductorio al uso de redes bayesianas en el tratamiento de datos de expresión genética para la obtención de redes genéticas. Contiene un repaso del trabajo realizado por otros autores, pero centrándose en las aportaciones realizadas por el autor y colaboradores y desde el punto de vista de la interpretación de los resultados biológicos.

Llegados a este punto, se hace necesario remarcar que los distintos trabajos que ahora se van a presentar tienen, en su mayoría, una importante contribución biológica, a la cual no vamos a hacer mención debido a que se escapa de los objetivos de esta memoria. Por tanto, nos vamos a centrar en comentar los distintos trabajos existentes en la obtención de redes genéticas usando redes bayesianas (estáticas o dinámicas) y en la clasificación de los datos de expresión genética usando clasificadores bayesianos.

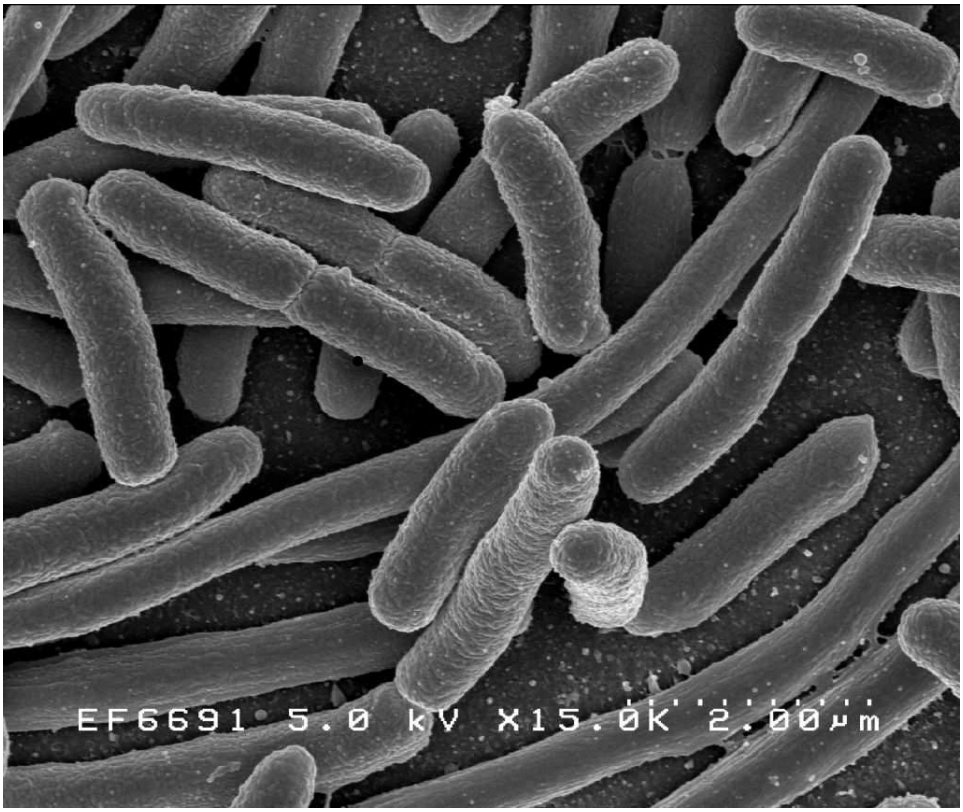


Figura 2.11: Foto de *Escherilia Coli*, quizás el organismo procariota más estudiado por el ser humano. Se trata de una bacteria que se encuentra generalmente en los intestinos de los animales (Fuente Wikimedia http://commons.wikimedia.org/wiki/File:EscherichiaColi_NIAID.jpg).

### 2.5.1. Uso de redes bayesianas para construir redes de regulación genética.

Los datos provenientes de microarrays de ADN tienen, como vimos, distintos problemas; no obstante, las redes bayesianas tienen ciertas características muy útiles para el análisis de este tipo de datos, como la habilidad de trabajar con datos perdidos, usar variables ocultas (como, por ejemplo, los niveles de proteína) que pueden tener un efecto en los niveles de expresión

de los genes. También pueden describir procesos locales, tienen la posibilidad de poder obtener conclusiones causales a partir de las redes obtenidas, son visualmente interpretables y la obtención de la red genética que explique el proceso fisiológico subyacente es inmediato. Además, las redes bayesianas pueden representar relaciones estocásticas entre genes (pues representan una distribución de probabilidad para los genes en la red), lo cual es especialmente interesante debido a que la expresión genética tiene una componente estocástica [228], y además son datos con gran cantidad de ruido [41].

Pero no todo son ventajas, los datos de expresión genética sufren de lo que se denomina la *maldición de la dimensionalidad* [30]: el número de casos es escaso -debido al coste de este tipo de experimentos- y el número de variables -genes-, suele ser bastante grande, en muchas ocasiones varios órdenes mayor que el número de casos. Debido a esto, es extremadamente difícil encontrar la red que mejor represente a esos datos e incluso distintas redes pueden explicar igual de bien dichos datos. Si nos fijamos sólo en el reducido número de muestras, podemos observar que se puede dar el caso en que las redes aprendidas detecten relaciones que no existen y debido al número de genes a estudiar, el coste computacional puede ser excesivo (recordemos que en el aprendizaje estructural de la red bayesiana el orden era super-exponencial en relación al número de nodos).

Por otro lado, la mayoría de métodos de aprendizaje de redes bayesianas trabajan con datos discretos y los datos de expresión genética son continuos, lo cual, hace que sea necesario discretizarlos provocando una pérdida de información. No obstante, la discretización es más estable frente a las variaciones aleatorias o sistemáticas que se producen en los niveles de expresión de los microarrays.

#### 2.5.1.1. Algoritmos de aprendizaje adaptados a datos de expresión genética.

El principal problema que nos encontramos para aprender redes bayesianas es, como hemos visto, la dimensionalidad del mismo o, simplemente, el gran número de variables disponibles. La aproximación más clásica para resolver este problema es la utilización de un algoritmo de selección de características (que en este tipo de datos se les denomina selección de genes [173]), para quedarnos con aquellas variables más relevantes y reducir

la dimensionalidad de los datos. No obstante, se han propuesto otro tipo de enfoques para superar este obstáculo: distintos autores han propuesto algoritmos de aprendizaje que les permitan aprender redes bayesianas adaptadas a este tipo de datos y que permitan trabajar con un gran número de variables.

Friedman y col. [117] presentan un algoritmo de aprendizaje orientado a bases de datos con muchas variables que posteriormente será usado en datos de microarrays de ADN. El método que presentan es un algoritmo iterativo que restringe los posibles padres de cada nodo a un pequeño conjunto de candidatos, aprenden una red con estos conjuntos de candidatos parra cada nodo y dicha red es utilizada para escoger los mejores candidatos en la siguiente iteración. Para cada nodo se construye una lista de candidatos usando medidas de dependencia y utilizando los datos y la red generada en la iteración anterior. Nos quedamos con los mejores candidatos teniendo en cuenta la idea de que dos variables con fuerte dependencia se encontrarán una cerca de la otra en la red. Una vez que tenemos un pequeño conjunto de candidatos hacemos una búsqueda para encontrar la red que mejor verifique esa restricción. Para esta búsqueda se usa un esquema métrica+búsqueda, utilizando la métrica descomponible *BDe* con una distribución *a priori* uniforme y con un tamaño muestral equivalente a 10. Se repite el proceso hasta que se deje de mejorar la métrica. El algoritmo de búsqueda usado es una versión simple de la búsqueda Tabú. Concluyen que este algoritmo es significativamente más rápido que otros métodos, sin perder calidad en las estructuras aprendidas[3].

Si en el anterior trabajo los autores buscaban un algoritmo de aprendizaje que pudiese trabajar con un gran número de variables, en [120] proponen un método para aprender redes más robustas a partir de pocos casos como ocurre en los datos de expresión genética, redes que puedan distinguir relaciones correctas del ruido. Para ello, utilizan *bootstrap* y para cada red generada en cada iteración, se fijan en una serie de patrones estructurales: enlaces no dirigidos en PDAGs [76], relaciones de orden en PDAGs, y mantos de Markov de una variable. Obtienen qué aspectos como un manto de Markov o una ordenación parcial de variables son más robustos que la existencia de enlaces no dirigidos en PDAGs. Estos aspectos deben ser a los que más

[3]El hecho de que se limite la búsqueda a un número determinado de padres por nodo, nos puede llevar a pensar que estamos perdiendo la posibilidad de encontrar redes que mejor describan a los datos, no obstante, recientemente se ha demostrado que las redes genéticas están poco conectadas, y el número medio de reguladores de un gen es menor que dos [215].

atención les prestemos a la hora de promediar la red a partir de las distintas redes obtenidas. En los experimentos utilizan, entre otros tipos de bases de datos, datos de expresión genética provenientes del ciclo celular de la levadura *Saccharomyces Cerevisiae*[4].

Combinando las ideas presentadas en los anteriores trabajos, en [122] Friedman y col. presentan las redes bayesianas como una herramienta prometedora para el análisis de datos de expresión genética, por varios motivos: primero, porque son especialmente útiles en la descripción de procesos donde hay componentes que interactúan de forma local; segundo, porque las bases estadísticas de las redes bayesianas y los algoritmos computacionales usados están bien asimilados y han sido usados con éxito en muchas aplicaciones; y tercero, la redes bayesianas proporcionan modelos de influencia causal. El algoritmo de aprendizaje es el que vimos antes [117]. También se usa bootstrap como en [120], pero en este caso, los patrones estructurales estudiados son el manto de Markov de las variables (ya que nos indican que dos genes están relacionados en algún proceso o interacción biológica) y las relaciones de orden (nos pueden indicar que un gen puede ser un ancestro causal de otro). Aplican este enfoque conjunto al estudio de datos de expresión genética provenientes del ciclo celular de la levadura *S. Cerevisiae* [319], sin utilizar ningún tipo de conocimiento biológico previo, ni restricciones y discretizando los datos en tres categorías (por debajo, igual o superior que el nivel de control, que se puede determinar experimentalmente o utilizando el valor medio).

Dado que los perfiles de datos de expresión genética de mutaciones proporcionan una gran variedad de medidas acerca de cómo la célula responde a las perturbaciones[5], en [265] se analizan estos datos y extienden el esquema presentado en el anterior trabajo [122] de cuatro formas: uno, adaptándolo para aprender con intervenciones [81] utilizando una métrica modificada que no es estructura-equivalente y que determina la dirección de la causalidad;

[4]Spellman y col. [319] hicieron público estos datos en http://cellcycle-www.stanford.edu/, siendo uno de los más utilizados en trabajos con datos de expresión genética dada su disponibilidad. Es un conjunto de datos de expresión genética, obtenidos a partir de microarrays de ADN, del ciclo celular de la levadura *Saccharomyces Cerevisiae*, tomados en distintos intervalos de tiempo.

[5]Los experimentos con perturbaciones son claves para obtener funciones de genes o caminos de regulación. Este tipo de experimentos son una técnica habitual donde se perturba la actividad de un gen de interés para estudiar los efectos producidos en la actividad de otros genes.

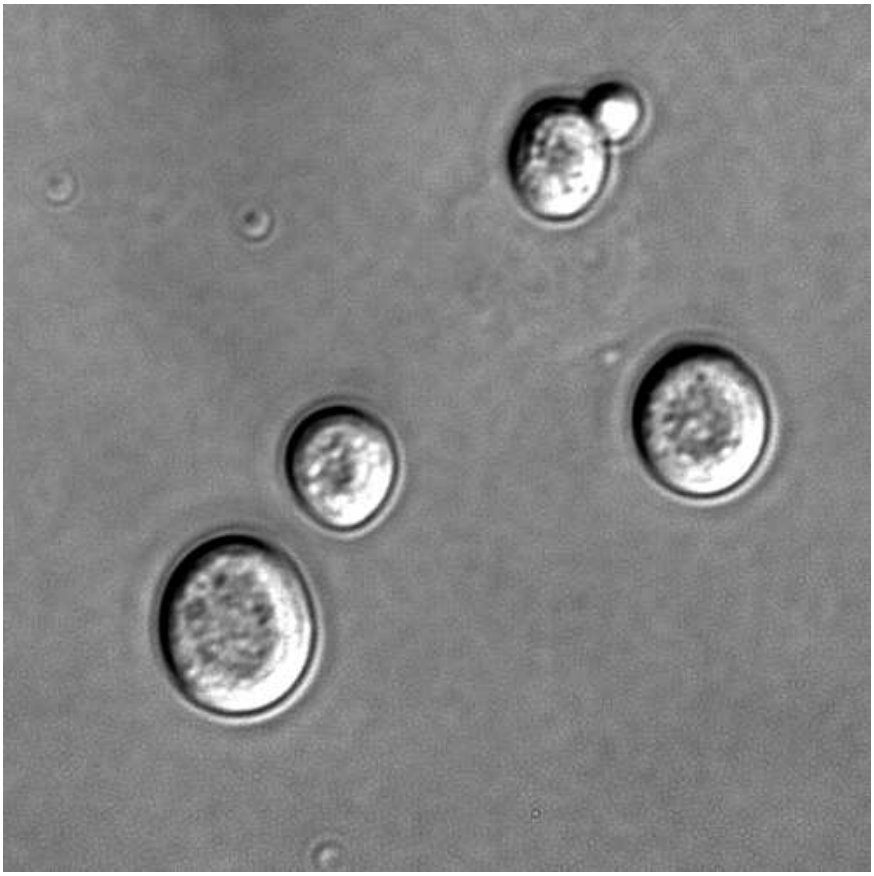

Figura 2.12: La levadura de cerveza (*Saccharomyces Cerevisiae*) es un hongo unicelular, un tipo de levadura utilizado industrialmente en la fabricación del pan, cerveza y vino (Fuente Wikimedia http://commons.wikimedia.org/wiki/File:S_cerevisiae_under_DIC_microscopy.jpg).

dos, modificando el esquema de discretización de los datos, se usa la técnica de agrupamiento K-medias para obtener los estados de discretización de cada gen en función de su actividad (variación en los estados de expresión); tres, aprenden y estudian nuevos aspectos utilizando *bootstrap*: mediador, activador e inhibidor, y, a su vez, explicando cómo se representan este tipo de relaciones entre genes con redes bayesianas; cuarto, describen cómo usar las características aprendidas (representan relaciones entre dos o tres genes) para construir subredes con un fuerte significado estadístico mediante dos

métodos (enfoque naïve y enfoque basado en la métrica). Aplican su estudio al análisis del compendio Rosetta de perfiles de expresión de la *S. Cerevisiae* [159]. Distinguen dos tipos de perturbaciones, un primer tipo que incluye perturbaciones de borrado de genes y sobreexpresión de datos genéticos. Ambas perturbaciones implican un cambio en la expresión de un gen mutante. Este tipo de perturbaciones los modelan en la métrica usada. El segundo tipo de perturbaciones son la sensibilidad a la temperatura y mutaciones genéticas, que tiene un efecto indirecto en el nivel de expresión. Para modelizar este tipo de perturbaciones utilizan *variables indicadoras* que serán variables sin padres en la red.

Friedman presente en los trabajos anteriores, también colabora en [104] donde se presenta un algoritmo especializado pero más enfocado al aprendizaje de redes bayesianas con variables ocultas. Este trabajo se basa en el algoritmo estructural EM [115] y el enfoque del cuello de botella de la información (*information bottleneck*) [339]. El algoritmo propuesto construye la estructura y parámetros de la red sin usar conocimiento previo y sin caer en óptimos locales. Lo prueban en distintos problemas, entre los que destacamos el conjunto de estrés de la levadura[6], discretizando los datos en tres estados y haciendo sólo entrenamiento y test.

Otros autores también han estudiado métodos para tratar con el problema del elevado número de variables que presentan los datos de expresión genética, como es el presentado por Peña y col. en [267, 269], donde utilizan la métrica BIC y el algoritmo de búsqueda KES (k-greedy equivalence) [246] y limitan el número de vecinos en la búsqueda, aplicando su propuesta a datos sintéticos y a datos de expresión genética [144] (discretizados en cuatro estados y con 320 casos y 32 genes). Extraen ciertos aspectos de interés y realizan estimaciones de la confianza en los aspectos obtenidos. Los aspectos de interés estudiados son: enlaces dirigidos y no dirigidos (reflejan interacciones inmediatas entre variables, los dirigidos además reflejan posibles relaciones causales), caminos dirigidos (establecen órdenes entre variables) y vecinos en el manto de Markov. La estimación de la confianza es la fracción de modelos que contienen dicho aspecto en los diferentes óptimos locales durante la fase

[6]Conjunto de datos de expresión genética de Sachs y col. [125], donde se mide la respuesta de la levadura *S. Cerevisiae* a diferentes condiciones ambientales. Es otro conjunto de datos bastante utilizado en la literatura que está disponible en http://genome-www.stanford.edu/yeast_stress/ .

aprendizaje.

Otro algoritmo distinto propuesto por los mismos autores lo encontramos en [268], donde presentan un algoritmo de aprendizaje (que denominan algoritmo GPC) basado en tests de independencias, capaz de tratar datos discretos y continuos. Para restringir el espacio de búsqueda estudian para un nodo semilla $S$ un entorno de radio $R$. Lo aplican a datos de expresión de genética de la levadura *S. Cerevisiae* (compendio Rosetta [159]) donde hay 300 casos con 6316 genes pero usando un gen de semilla y un radio $R = 2$.

Huang y col., en [158], también proponen un algoritmo que aprende una red bayesiana con muchas variables. Sigue la idea de los algoritmos basados en tests de independencias, no obstante, usa la información mutua condicionada a modo de test. El resultado es una red que denominan *basta* y que no refleja exactamente las relaciones de dependencia pero que les da una idea de las relaciones subyacentes. Hay que destacar que en el conjunto de datos utilizado [319], hacen una selección de características previa, quedándose con 76 genes.

Con una idea inicial parecida a la del anterior trabajo, Wang y col. en [349] presentan un algoritmo híbrido donde, primero se realiza un aprendizaje basado en tests de independencias que es una variación del algoritmo PC que usa heurísticas para disminuir el número de tests, y después, la red obtenida es utilizada como punto de inicio para un proceso de aprendizaje métrica+búsqueda (métrica BDeu y búsqueda MCMC). Lo aplican con éxito a problemas con un gran número de variables, por ejemplo, usan 800 genes de [319]. Lo compara con otros algoritmos similares como el de Friedman y col [117] o el de Peña y col. [268], aduciendo que en su propuesta no es necesario usar como parámetro el número de padres/vecinos por nodo.

#### 2.5.1.2. Utilización de conocimiento adicional.

No sólo mediante algoritmos especializados se ha intentado superar la maldición de la dimensionalidad de los datos de expresión genética. Algunos autores han propuesto que ante la escasez de muestras de este tipo de datos, es conveniente utilizar conocimiento experto o utilizar cualquier otro conocimiento adicional que esté a su alcance. A este tipo de conocimiento se le suele denominar *conocimiento previo*, pues es usado al principio de los

algoritmos de aprendizaje de redes bayesianas.

En [306], Segal y col. por un lado construyen una estructura naïve bayes de gaussianas para datos de expresión genética y, por otro lado, construyen una red de Markov para datos de interacciones proteína-proteína. Se unen las dos redes, aprendiendo los parámetros del modelo unificado mediante el algoritmo EM. De esta forma y utilizando un conjunto de datos adicional se construye de forma rápida la estructura de la red, aunque sea con una topología bastante limitada.

Gifford en [134] propone utilizar modelos gráficos para datos de expresión genética como una herramienta libre de parámetros capaz de modelizar redes genéticas. Como aplicación Hartemink, Gifford y col. en [142, 143] construyen a mano dos redes bayesianas rivales que modelizan el metabolismo de la galactosa en la levadura *S. Cerevisiae* y usan una métrica bayesiana para comparar dichos modelos. Posteriormente, cogen tres de los genes usados en los anteriores modelos y calculan todas las posibles redes entre dichos genes y calculan su valor de la métrica. Obtienen que cuando existe un determinado enlace mejora la métrica y cuando existe otro empeora, ambos resultados son acordes con los conocimientos biológicos del problema. También presentan lo que denominan enlaces etiquetados con el objetivo de representar conocimiento biológico adicional en diversas etapas de refinamiento, de forma que en el enlace se indique cuando un gen regula positiva o negativamente a otro, o indique simplemente que hay una relación entre ambos genes. Modifican la métrica para que tenga en cuenta los enlaces anotados, y calculan varias extensiones de las redes bayesianas añadiendo anotaciones a los enlaces, calculan el valor de la métrica y obtienen que enlaces anotados que no satisfacen los datos provocan peores valores de la métrica aún cuando la estructura subyacente es correcta. Por ello, presentan los enlaces anotados como un discriminador de tipos de relaciones presentes los datos.

Los mismos autores presentan en [144] un método para la obtención de redes de regulación genética a partir de datos de expresión genética y datos provenientes del análisis de localización genética [287]. Estos últimos son utilizados para guiar la construcción de la red bayesiana. Utilizan un enfoque métrica+búsqueda, donde el algoritmo de búsqueda es enfriamiento simulado. Para evitar sobreajuste y dar robustez al proceso promedian los enlaces obtenidos. Obtienen que el modelo sin restricciones no encuentra por sí solo algunos enlaces que deben de estar. De esta forma, nos dicen, la utilización

de datos de localización genética es un buen complemento a los datos de expresión genética.

Zhu y col usan datos genotípicos en [373], y en [374]. Además, usan interacciones proteína-proteína y datos de sitios de unión de los factores de transcripción[7] . Dicha información previa es usada como evidencia a priori en el algoritmo de aprendizaje, utilizando como método de búsqueda MCMC y métrica BIC. Para evitar el sobreajuste a los datos, además, generan mil redes y hacen un promedio de las mejores.

Utilizando datos de interacciones proteína-proteína e información previa a partir de la literatura existente, Djebbari y Quackenbush en [98] crean una red con dicho conocimiento que será usada para inicializar un algoritmo de búsqueda local que además utiliza una métrica BDe. Siguiendo la idea propuesta de Friedman y col. [122], realizan bootstrap promediando la red obtenida con aquellos enlaces que aparezcan más del 70 % de las veces.

En cambio, en [214] Philip y col. no utilizan información de otros conjuntos de datos sino que se basan en una red construida por un experto basándose en la literatura existente. Utilizando como algoritmo de búsqueda MCMC y métrica MDL, utilizan restricciones duras de ausencia y existencia en base a la red dada, es decir, limitan el espacio de búsqueda con enlaces que obligatoriamente deben de estar en las redes candidatas y aquellos que están prohibidos. La red obtenida por el experto es usada para generar el conjunto de datos sintéticos sobre los cuales probarán su método.

Con un planteamiento similar, Almasri y col. en [11] utilizan una red genética construida previamente en el trabajo de otro autor [210]. Siguen tres esquemas para aprender la red bayesiana: en el primer caso, generan redes aleatoriamente y las miden con la métrica K2; en el segundo caso utilizan la estructura de la red conocida como punto de partida para el algoritmo de aprendizaje; en el tercer caso, restringen el uso de posibles enlaces en el espacio de búsqueda como se hace en [321], utilizando la red dada por el experto.

[7] Un factor de transcripción es una proteína que participa en la regulación de la transcripción del ADN. Al ser activados adquieren la capacidad de regular la expresión genética, bien activando, bien reprimiendo la transcripción de diversos genes.

#### 2.5.1.3. Trabajando con datos continuos.

Otro de los problemas que vimos que presentaban los datos provenientes de microarrays de ADN era el hecho de que los niveles de expresión se movían en un rango de valores continuos, mientras que la mayoría de los métodos que trabajan con redes bayesianas trabajan con datos discretos; motivo por el cual se hace necesaria su discretización, siendo la discretización en tres estados la opción más aceptada (propuesta inicialmente en [122]).

No obstante, la discretización de los datos produce una pérdida de información por lo que Imoto y col. [166] utilizan modelos de regresión no paramétricos aditivos ($\beta$-splines) con el objetivo de poder captar dependencias entre genes que no sean sólo lineales (como captan los modelos de gaussianas). Para este tipo de redes bayesianas presentan una métrica para este tipo de datos, y basada en medidas de información, que es la BNRC (Bayesian Nonparametric Regression Criterion) y como esquema de búsqueda una variación de la búsqueda local (backfitting [145]). Hacen una simulación Monte Carlo para probar su método y también lo aplican a la base de datos de *S. Cerevisiae* [319] con el objeto de comparar su método con el de Friedman y col [122], obteniendo que sus resultados se aproximan bastante a éstos pero además obtienen dependencias no lineales.

Usando también regresión no paramétrica y la métrica BNRC, en [249] proponen un variación del algoritmo de búsqueda K2, que denominan HN, donde no es necesario una ordenación previa de las variables.

Bastos y col en [28] también utilizan un modelo de regresión no paramétrica ($\beta$-splines), con una búsqueda local donde restringen el número de padres por nodo, pero en este caso utilizan la métrica BIC, en contraposición a la métrica BNRC aduciendo que es más fácil y rápida de calcular. Repiten el experimento varias veces quedándose con aquellos enlaces que se repiten un número de veces mayor que un umbral. En [27] mejoran el método usando una red construida por un experto, a partir de dicha red fuerzan algunos enlaces en el proceso de búsqueda.

Ott, Imoto y col. en [256] usan como algoritmo de búsqueda programación dinámica y distintas métricas (BRNC, BDE y MDL). Proponen que limitando el número de padres por nodo a un conjunto limitado, el proceso de búsqueda es exponencial (en lugar de super-exponencial), pudiendo en-

contrar así redes óptimas para la métrica usada. Limitando el número de padres a seis, obtienen redes óptimas de 20-30 genes con la métrica BNRC de 20-40 genes para las métricas MDL y BDe. Prosiguen en [257] utilizando sólo la métrica BNRC y dividen los genes en clústeres con la intención de encontrar las redes óptimas para dichos clústeres con el enfoque propuesto en su anterior trabajo. De dichos clústeres obtienen el conjunto de padres, para construir los clústeres utilizan dos métodos: uno sin información adicional y otro con información biológica. En [255] obtienen un conjunto de redes óptimas (con el mismo valor de la métrica) para un mismo conjunto de datos y hacen promedio de las mismas, con el objetivo de identificar patrones en las redes.

En [172, 170] Imoto y col. proponen una mejora a sus redes con regresión no paramétrica basados en $\beta$-splines [166], construyendo redes bayesianas con modelos de regresión no paramétricos con varianzas de error heterogéneas (regresión no paramétrica heterocedástica), siendo más resistente a medidas atípicas o erróneas que [166]. Para construir la red bayesiana utilizan una búsqueda local y presentan la métrica BNRChetero (Bayesian Nonparametric Heterocedastic Regression Criterion), la cual, es una máetrica basada en medidas de información y trabaja con el enfoque continuo que presentan . En la búsqueda local crean una matriz cuadrada donde se almacena el valor BNRChetero entre cada par de genes y en base a este valor se realiza la búsqueda (se reduce la búsqueda en el número posible de padres). Concluyen que su método extrae satisfactoriamente información que puede verse en forma visual (red bayesiana) y que está de acuerdo con conocimiento biológico que se tiene. Utilizando la anterior metodología, en [171], aplican el método *bootstrap* para estudiar los enlaces: calculando la intensidad del enlace y grado de confianza en la causalidad bayesiana. Construyen una red donde se reflejan las intensidades de cada enlace y eliminan aquellos enlaces que están por debajo de un umbral. En [178] usan *bootstrap multiescala*, una variación del bootstrap donde cambia el calculo de la intensidad de cada enlace.

En [334, 333] Tamada, Imoto y col. proponen usar además información de la secuencia de ADN para construir la red bayesiana, basándose en la idea de que si un gen padre es un factor de transcripción sus genes hijos pueden compartir una misma secuencia de consenso en sus regiones promotoras de la secuencia de ADN. Construyen una red bayesiana usando la metodología de [166, 170]. A partir de esta red detectan genes que pueden considerar-

se candidatos a factores de transcripción y definen conjuntos de genes que puedan ser corregulados para cada candidato. Utilizan un método de detección de secuencias de consenso [21, 22] para cada conjunto de posibles genes corregulados. Una vez encontrados las secuencias de consenso usan esta información (como en [168]) para construir de nuevo la red bayesiana. Repiten este proceso hasta que la estructura de la red no cambia considerablemente. Evalúan su método con una simulación Monte Carlo y lo aplican al estudio de *S. Cerevisiae* centrándose en aquellos factores de transcripción que en [166], regulaban a muchos genes. Concluyen que obtienen mediante este método redes bayesianas más precisas y que además detectan elementos promotores.

Imoto y col. en [169, 168] proponen utilizar más fuentes de información adicionales usando conocimiento biológico de diversas fuentes: interacciones proteína-proteína, interacciones ADN-proteína, información de sitios de localización genética y trabajo previo. Para construir la red bayesiana siguen el esquema de [170] modificando la métrica BNRChetero de forma que pueden incorporar conocimiento previo dentro de la distribución *a priori*. Para validar su método utilizan una simulación Monte Carlo y lo aplican a análisis de datos de *S. Cerevisiae*. Construyen una red que modeliza el conocimiento previo. Concluyen que al utilizar información biológica se extrae más información de los microarrays y se estima la red genética de forma más precisa. En [353, 352] hacen una variación del trabajo [168] pero usando MCMC en el aprendizaje estructural y paramétrico, utilizando información adicional de diversas fuentes.

En [299] Savoie, Imoto y col. utilizan redes booleanas y bayesianas (siguiendo la metodología de [172]) para estudiar las dianas moleculares y las cascadas de genes reguladoras[8] afectadas por el antimicótico Griseofulvin (antihongos). Para ello obtienen una serie de microarrays de hongos expuestos al antibiótico. A partir de las dos redes construidas (booleana y bayesiana) obtienen las cascadas de genes afectadas por el Griseofulvin. Concluyen que las dos técnicas utilizadas demuestran la utilidad de usar datos de expresión genética y redes genéticas para determinar los mecanismos de actuación de un compuesto.

[8] Una *cascada de genes*, es un conjunto de genes que se expresan secuencialmente, donde la expresión de un gen provoca la expresión del siguiente.

En la misma línea, en [332] buscan encontrar genes afectados por medicamentos, mediante una red aprendida a partir de datos de expresión genética y de la respuesta al medicamento, los cuales están tomados en intervalos de tiempo. Aprenden una red bayesiana mediante regresión no paramétrica, la discretizan para inferencia (en tres estados cada variable), y de la red infieren los genes afectados por los medicamentos. Para ello realizan la inferencia con las muestras de los efectos del medicamento.

En [242] se limitan a usar como información adicional interacciones proteína-proteína. En la red bayesiana utilizan un modelo aditivo de regresión no paramétrica [166, 170] usando la métrica BNRC pero incorporando la información de las interacciones proteína-proteína como en [168]. Cuando un gen está regulado por un complejo proteínico consideran dicho complejo padre del gen. Los genes que forman parte de un mismo complejo proteínico tendrán sus datos de expresión genética fuertemente correlacionados, por ello modelizan los complejos proteínicos como nodos en la red bayesiana. Para construir estas variables usan análisis de componentes principales. Evalúan su método con datos de expresión genética de *S. Cerevisiae* usando sólo 99 genes y comparando el resultado con una red genética extraída de la base datos KEGG[9]. Obtienen que la red de regulación genética es más precisa, al usar datos de interacciones proteína-proteína y datos de expresión genética a la vez que modelizan complejos proteínicos. En [243] además de generar la red genética, generan una red de interacciones proteína-proteína.

También con la intención de trabajar con datos continuos pero con un planteamiento distinto, Ko y col. en [192] construyen las redes mediante una búsqueda voraz y la métrica BIC, limitando el número de padre por nodo a 2 y 3. Usan mixturas de gaussianas y el algoritmo EM para obtener los parámetros de las mixturas.

#### 2.5.1.4. Redes modulares y clustering.

La modularidad y el ordenamiento jerárquico son propiedades inherentes a los sistemas biológicos [284, 25], por tanto, resulta en cierto modo natural ordenar los genes en clústeres o módulos, y utilizar algoritmos de aprendizaje de forma local en dichos conjuntos de genes. No obstante, muchos métodos

[9]La base de datos KEGG (Kyoto Encyclopedia of Genes and Genomes) [179] es una de las bases de datos de referencia para redes genéticas de distintas especies.

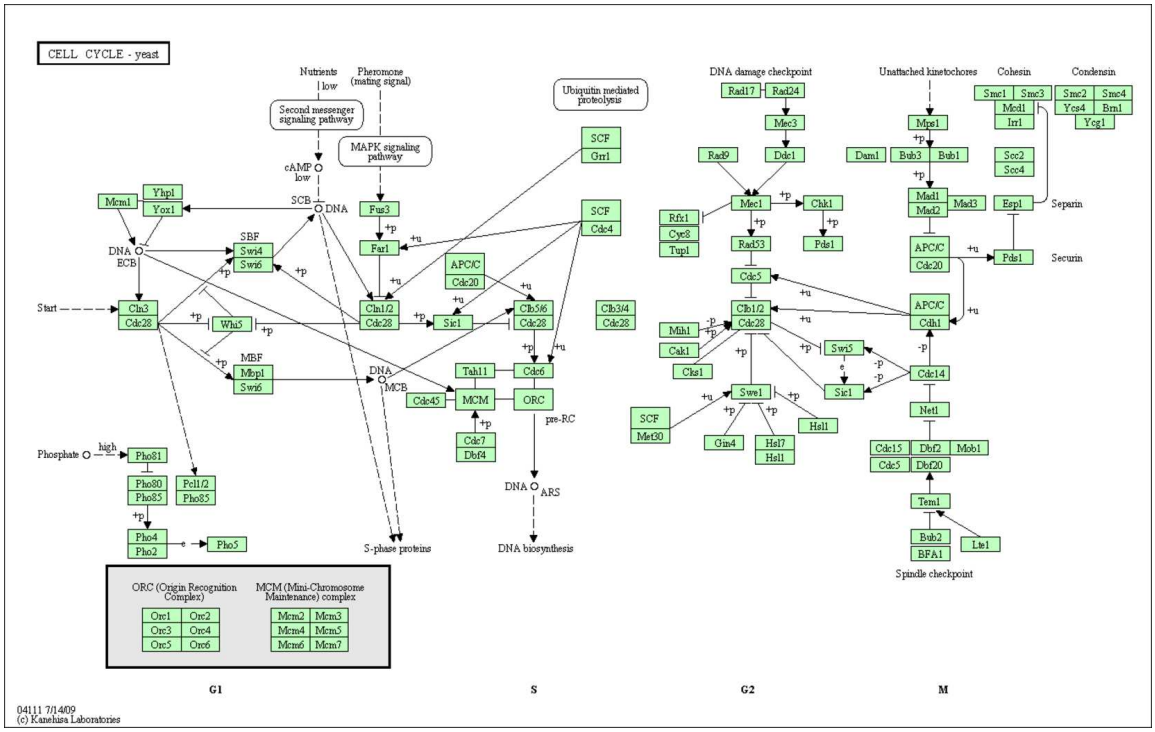


Figura 2.13: Diagrama del ciclo celular de la levadura *Saccharomyces Cerevisiae*, extraída de la base de datos KEGG.

que usan clustering o aprendizaje en redes locales, lo que buscan en realidad es acortar la diferencia entre el número de genes y el número de casos.

Segal y col. [303, 304] presentan las redes bayesianas modulares, donde se agrupan las variables en módulos. Definen un módulo regulador al conjunto de genes que se regulan conjuntamente. Las variables que pertenecen a un mismo módulo comparten padres y parámetros. Presentan una métrica bayesiana (basada en [148]) para aprender redes de este tipo y un algoritmo iterativo compuesto de dos pasos basados en búsqueda local: en un primer paso, se realiza una búsqueda de la estructura entre módulos y, en un segundo paso, se lleva a cabo la asignación de variables a cada uno de los módulos. Aplican su propuesta a datos bursátiles, sintéticos y de expresión genética (levadura -estrés). En estos últimos limitan el número de padres por módulo, pues conocen (información previa) qué genes regulan al resto, por tanto, en

cada módulo tendremos genes corregulados.

Hacen una variación de su anterior modelo en [305], donde el algoritmo toma como entrada, datos de expresión genética y un conjunto precompilado de genes candidatos a ser genes reguladores Para aprender la red usan un algoritmo de aprendizaje iterativo siguiendo la esencia del algoritmo EM estructural [115]. En cada iteración se tienen el paso E y el paso M: en el paso M se empieza con una partición de los genes en módulos y aprende el mejor *programa de regulación* para cada módulo usando árboles de regresión. En el paso E, dados los *programas de regulación* se determinan que módulos de los asociados a un *programa de regulación* predice mejor el comportamiento de cada gen. Su método identifica módulos de genes corregulados, sus reguladores y las condiciones bajo las cuales ocurre dicha regulación, generando hipótesis acerca del papel físico de un regulador, los genes que regula y las condiciones en las que ocurre (hipótesis del tipo *el gen regulador $X$ regula el módulo $Y$ bajo las condiciones $W$*). Algunas de estas hipótesis son demostradas, sugiriendo acciones de regulación para proteínas sin caracterizar previamente.

Usando las redes modulares de Segal, Novershtern y col. en [248] integran información de diversas fuentes y conjuntos de datos públicos y heterogéneos para la caracterización genética del asma. Afirman que obtienen conclusiones que no habría sido fácil de obtener utilizando esos datos por separado.

En [233] se presenta una continuación del trabajo de Segal en redes modulares. Michoel y col. proponen LeMoNE, una herramienta para generar redes modulares. Además de probarlo en el conjnunto de datos levadura-estrés, utilizan el sistema SynTReN que permite generar bases de datos sintéticas de expresión genética.

Si bien las redes modulares pueden considerarse clústeres de genes, en [64] Chang y col. utilizan de forma explícita clústeres de genes para reducir la dimensionalidad del problema en lo que denominan abstracción de atributos (*Feature abstraction*). Estudian la relación de datos de expresión genética provenientes de muestras de células humanas cancerosas y la actividad de medicamentos. Usan la métrica BD y una búsqueda voraz para construir la red sobre las abstracciones de atributos.

Liu y col. también construyen clústeres de genes para reducir el número de variables en [364], pero además con el objetivo de encontrar genes con

funciones biológicas relacionadas. Dentro de cada clúster aprenden una red utilizando la métrica MDL y un algoritmo bioinspirado de búsqueda (un algoritmo evolutivo basado en el sistema inmune).

Con el objetivo de agrupar genes comunes, Peña y col. en [271] usan EDAs [208] para hacer aprendizaje no supervisado de redes bayesianas, aplicando los métodos estudiados a datos sintéticos y a la base de datos de [136], discretizando, en este caso, los datos a tres estados. El clustering de la red bayesiana lo hacen usando el EDA dentro del algoritmo bayesiano estructural EM [115] (*BSEM-UMDA*) y, directamente, usando sólo el EDA (*UMDA*). Concluyen que ambos algoritmos son válidos para detectar agrupamientos de genes, aunque BSEM-UMDA escala mejor y que la validación de los agrupamientos de genes obtenidos sugieren que puedan tener sentido biológico.

Por otro lado, en [372] se genera un conjunto de redes aleatorias y las mejoran mediante búsqueda local. Zhu y col. se quedan con un promedio de las mejores asignando un factor de confianza a cada enlace. Los genes son agrupados en cinco clústeres y además de los datos de expresión genética obtienen variables con información de los experimentos (estudian la respuesta de la levadura al ácido nítrico). Ven su propuesta como un proceso iterativo, puesto que cada experimento es usado para comprender mejor lo que está pasando y diseñar los siguientes experimentos.

En [26] Barash y col. también utilizan clústeres pero no con el objetivo de reducir el número de variables, ya que describen una técnica de aprendizaje no supervisado que denominan agrupamiento CSI y que presentan como una versión mejorada del naïve bayes selectivo no supervisado, pero capaz de representar independencias dependientes del contexto. Aprenden estos agrupamientos CSI utilizando un enfoque bayesiano basado en el algoritmo EM estructural [115]. Los datos sobre los cuales aplican su técnica de agrupamiento son de expresión genética y sobre datos de localización de sitios de unión. Para evaluar su modelo utilizan datos sintéticos y datos reales [319, 125]. En estos últimos utilizan una inicialización de los agrupamientos basados en el K-medias. Concluyen diciendo que su método permite combinar datos genéticos y genómicos y les permite identificar y caracterizar agrupamientos de genes con un comportamiento coherente, aunque los resultados se ven afectados por la inicialización.

Un trabajo relacionado es el de Sen y col. en [307]. Definen lo que son los contextos celulares (por ejemplo, la expresión de una célula cancerígena no puede ser igual a una sana) y proponen usar clústeres para poder modelizar dichos contextos con datos de expresión genética. Para ello crean un conjunto de cinco clústeres y posteriormente crean una red bayesiana para cada clúster. En este caso los clústeres no se construyen sobre las variables, se hacen sobre los casos, pues argumentan que una misma variable se puede expresar en dos contextos celulares distintos. De esta forma también representan independencias dependientes del contexto.

#### 2.5.1.5. Selección de características para obtener redes genéticas.

Si bien comentamos que en la literatura existente se ha preferido usar otras vías (algoritmos especializados, clustering, información adicional) a la selección de características para reducir la dimensionalidad de los datos de expresión genética, algunos autores se han inclinado por esta opción más clásica. Es conveniente mencionar que en la mayoría de los trabajos se hace una selección de genes aunque no muy severa.

Quizás utilizar estos enfoques más clásicos no sea lo más adecuado para datos de expresión genética tal y como se puede ver en [247], donde el autor compara seis algoritmos de aprendizaje estructural clásicos de redes bayesianas para inferir redes de regulación genética y realiza una selección de atributos basándose en trabajos de otros autores, obteniendo resultados bastante malos.

Peeling y Tucker en [264] forman lo que denominan una red de consenso, uniendo redes obtenidas de conjuntos de datos distintos y que no se pueden combinar entre sí. Para hacer la unión de las distintas redes utiliza el PDAG equivalente para cada red y busca relaciones comunes en las distintas redes. La selección de características se basa en artículos previos.

Zhou y col. [371] también se basan en la selección de genes de otro trabajo previo, utilizan un métrica bayesiana y como algoritmo de búsqueda una variación del MCMC [370].

Realizando la selección de genes por un experto, Gamberoni y col. en [123] trabajan con el conjunto de leucemia y se quedan con dos conjuntos

de datos de 33 y 10 genes (el original posee 12558 genes). Posteriormente utilizan el algoritmo K2 y una variación que proponen del mismo K2lift que realiza una disminución del número de padres a estudiar por nodo, realizando varias iteraciones con distintos órdenes de variables.

### 2.5.2. Uso de redes bayesianas dinámicas para construir redes de regulación genética.

Las redes biológicas tienen ciclos y bucles de retroalimentación, sin embargo, una de las limitaciones de las redes bayesianas para modelar redes genéticas es que usan un grafo dirigido acíclico, por tanto, no pueden representar ciclos, algo que sí puede suceder en una red genética. No obstante, como vimos, las redes bayesianas dinámicas sí que pueden representar ciclos mediante los arcos temporales tal y como se puede ver en la figura 2.14. Por tanto, las redes bayesianas dinámicas nos van a permitir representar auto regulación y, otra característica importante, regulación en el tiempo.

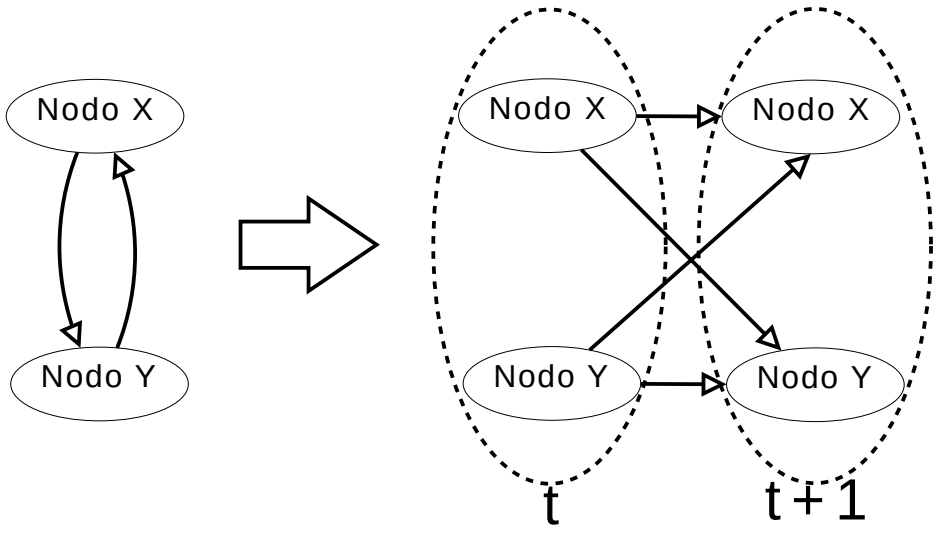


Figura 2.14: Un ciclo en una red bayesiana se puede ver como un grafo acíclico en una red bayesiana dinámica.

Otra ventaja de las redes bayesianas dinámicas es que, desde un punto

de vista teórico [253], es mejor tener pocos experimentos con muchas series temporales que muchos experimentos: para determinar causalidad, experimentos con dos series temporales son mejores que un experimento individual independiente.

Para introducirnos en el análisis temporal de datos de expresión genética en [24] encontramos un interesante artículo realizado por Bar-Joseph, que presenta un repaso de la investigación realizada sobre series temporales de datos de expresión genética, conteniendo una pequeña revisión del trabajo realizado en redes bayesianas dinámicas.

El primer trabajo donde se presentan las redes bayesianas dinámicas con el objetivo de aplicarlas a datos de expresión genética es el de Murphy y col. [240]. En él se explica en profundidad lo que son las redes bayesianas dinámicas y se presenta una revisión de técnicas de aprendizaje de las mismas, no obstante, estos modelos no se aplican a datos de expresión genética porque no encontraron bases de datos públicamente disponibles. En [348], Wang y col. se basan en la propuesta de Murphy y col. para aprender las redes pero además utilizan un sistema de clustering difuso [37] para reducir la dimensionalidad del problema, pasando de 281 genes a 23 clústeres. Sobre cada clúster aprende una red bayesiana.

Al igual que pasaba en las redes bayesianas estáticas la maldición de la dimensionalidad fuerza a los autores a utilizar algoritmos especializados e incorporar información adicional. Por ejemplo, Ong y col. en [251] utilizan un algoritmo especializado basado en el uso de mapas de operones [10] y datos de expresión genética. Motivados por la idea de que las células continuamente reprograman sus redes de expresión genética a lo largo del ciclo celular, utilizan redes bayesianas dinámicas para modelizar esta variación en el tiempo, aplicándolas a los cambios fisiológicos que afectan al metabolismo del triptófano[11] en la bacteria *E. Coli* [184]. Discretizan los datos a dos estados.

[10] En organismos procariotas hay muchos conjuntos de genes contiguos que se sabe que se transcriben a la vez y por tanto están fuertemente relacionados. Estas secuencias de genes que se transcriben juntos se denominan *operón*. En el mapa de un operón se tienen los genes que lo forman.

[11] El triptófano es un aminoácido esencial en la nutrición humana. Por ejemplo, es fundamental para promover la liberación del neurotransmisor serotonina, involucrado en la regulación del sueño

Antes de construir la red bayesiana dinámica construyen una red de regulación genética (BNreg). También construyen otra red bayesiana (BNop) a partir del mapa de operones usando naïve bayes. En esta red se muestran relaciones de causalidad entre genes y operones, con arcos de cada operón a sus genes asociados. En la distribución *a priori* utilizan conocimiento de un experto. Para construir la red bayesiana dinámica inicial replican para cada serie temporal la red BNop. En el aprendizaje de la estructura se fijan en genes u operones clave que son afectados por la ausencia/presencia de triptófano. Para cada uno de estos genes clave en BNreg (operones clave en el caso de BNop) consideran al resto de genes como posibles padres. En cada paso utilizan el algoritmo EM para actualizar las probabilidades y obtienen el logaritmo de la verosimilitud máxima que se utilizará para obtener los padres más probables. Concluye que la utilización de operones proporciona una mejor visión del metabolismo del triptófano y usándolos con redes bayesianas dinámicas nos proporciona una herramienta capaz de identificar operones en la *E. Coli* que se regulan de forma común. En [252] utilizan un mapa de operones más completo para generar una red bayesiana cuya estructura se repetirá a lo largo de intervalos de la red bayesiana dinámica (aquí no se usa BNreg) .

Husmeier propone la utilización de redes bayesianas como herramienta de ingeniería inversa en la obtención de redes genéticas en [161]. En [162], además de usar datos sintéticos, hace experimentos en datos reales usando el modelo propuesto por [366] y discretizando los datos en tres estados. En los experimentos prueba diferentes distribuciones *a priori* en la inferencia de la red bayesiana, utilizando más o menos información previa y concluye que los resultados dependen de las distribuciones *a priori* que se utilicen, mejorando cuanta más información se utilice.

Utilizando también información adicional en la distribución *a priori* del algoritmo de aprendizaje, Bernard y Hartemink [35] realizan una extensión al trabajo presentado en [144], usando ahora modelos dinámicos para construir redes de regulación genética dinámicas, a partir de datos de expresión genética y datos de localización de sitios de unión de los factores de transcripción. En la red utilizan una variable para explicar cómo el proceso regulador de los genes depende de la fase celular. Utilizan esta solución para no tener variables ocultas. Para aprender la estructura de la red utilizan el mismo esquema de [144]. Aplican su estudio a datos sintéticos y al ciclo celular de la

levadura (obteniendo los datos de expresión genética de [319] y los datos de localización de sitios de unión de [216]), discretizados en tres estados. Para evaluar la red aprendida la comparan con un modelo de red genética hecha a mano usando información experta de diferentes fuentes. Concluyen que la utilización de datos de localización de sitios de unión junto con datos de expresión genética incrementan la precisión de las redes y reduce la cantidad (de por sí escasa) de datos de expresión genética necesarios.

Los factores de transcripción son también utilizados por Zou y Conzen [375] como información adicional a su modelo. En su propuesta, limitan los posibles genes reguladores consiguiendo así reducir el espacio de búsqueda e incrementar la precisión de la red obtenida. Para la selección de los posibles reguladores buscan el punto en el tiempo en el que los diferentes genes cambian su expresión (se sobreexpresan o se infraexpresan). Un gen se considera un posible regulador de un gen diana si su cambio de expresión antecede en el tiempo al cambio de expresión de dicho gen diana. Posteriormente se calcula el retraso entre el cambio de expresión del posible regulador y del gen diana. Para cada grupo de posibles reguladores se generan todos los subconjuntos de ese grupo y para cada subconjunto, usando el retraso calculado, se organizan los datos de expresión de los posibles reguladores y sus dianas en matrices, calculan la probabilidad condicional para cada gen diana en relación a sus reguladores basándose en las matrices. Se calcula la métrica usando esas probabilidades condicionadas. Para cada gen diana, se selecciona el subconjunto de reguladores que tienen un valor mayor de la métrica. Estudian el caso en que se conozcan los factores de transcripción pero no sus genes dianas (información previa) y el caso en que no se dé ninguna información previa. En ambos casos, concluyen que con su método se obtienen mejores redes y en menos tiempo.

Zhang y col. [368, 369] usan redes bayesianas dinámicas también con conjuntos de datos de diferentes fuentes. En un primer paso construyen la red bayesiana dinámica sólo con datos de expresión genética discretizados en tres estados y utilizando el algoritmo EM estructural puesto que maneja datos perdidos. En un segundo paso utilizan como punto de partida la red anterior y ahora, además de usar datos de expresión genética, usan datos de localización.

Basándose también en el algoritmo EM estructural para redes bayesianas

dinámicas, Tienda-Luna y col. en [337] proponen una variación del mismo, denominado VBEM (del inglés, *variational Bayesian EM*) que se puede considerar una versión probabilística del algoritmo EM estructural y que aprende estructura y parámetros, además no es demasiado complejo así que lo ven útil para aprender redes con muchas variables. Para aumentar la robustez del modelo utilizan bootstrap aplicado a series temporales. Aplican su método a los datos de la levadura, comparando sus resultados con la base de datos KEGG. En [338] tenemos el mismo esquema pero enfocado al parásito de la malaria.

Como vimos, la discretización de los datos de expresión genética provocaba una pérdida de información. En la construcción de redes bayesianas dinámicas también se proponen modelos que nos permitan trabajar con datos continuos. Por ejemplo, Perrin y col. en [274] utilizan redes bayesianas dinámicas con variables continuas usando distribuciones gaussianas. Se centran en la red genética de la reparación S.O.S. del ADN[12] de la bacteria *Escherilia Coli* [291]. Utilizan lo que denominan *variables perdidas* para aquellos genes que son importantes en la red estudiada pero que para los cuales no tienen datos de expresión. Utilizan una variación del algoritmo EM ya que les permite inferir variables ocultas.

Huang y col. en [157] también usan un modelo lineal (gaussianas), pero para la búsqueda usan una variación del MCMC. En cambio, Grzegorczyk y col. en [140] proponen un modelo lineal y no homogéneo para representar datos continuos: mixturas de gaussianas. Modifican la métrica BGe [126] para su enfoque y utilizan para el aprendizaje estructural la búsqueda MCMC.

Trabajando también con datos continuos, pero basándose en regresión no paramétrica basados en $\beta$-splines, Kim, Imoto y col. en [188, 189, 187] utilizan redes bayesianas dinámicas basándose en el modelo presentado en [172]. Para ello definen una nueva métrica BNRCdynamic (dynamic BNRC). Aplican este paradigma al análisis de datos de expresión genética del ciclo celular de *S. Cerevisiae* [319], obteniendo que su método es una extensión del de [172] ya que soporta series temporales. En [186, 188] además comparan la red encontrada con una obtenida de KEGG. En [235] comparan las redes bayesianas dinámicas basadas en regresión no paramétrica con modelos de

[12]Esta red genética es responsable de la reparación del ADN cuando se produce un daño.

espacio de estado. En [167] también usan la información adicional utilizada en [169], pero como novedad, proponen un método tolerante a fallos en las base de datos usadas como información adicional: construyen la red, analizan la red resultante e intentan añadir arcos que están en la información previa mientras no empeore la métrica utilizada.

Un trabajo relacionado es el de Tamada, Imoto y col. [331] donde utilizan la metodología de [188] e información evolutiva. Usan el criterio de información evolutiva como el conjunto de pares de genes de dos organismos distintos. Aplican su método al estudio de *S. Cerevisiae* [319] y al *Homo Sapiens*, construyendo dos redes, cada una con datos de expresión genética del organismo respectivo y la información evolutiva de ambos organismos. Comparan las redes obtenidas para ambos organismos con las de KEGG, obteniendo que usar dicha información evolutiva adicional proporciona redes genéticas más precisas que usando sólo datos de expresión genética.

### 2.5.3. Uso de clasificadores bayesianos.

Dentro del uso de clasificadores bayesianos para el análisis de datos de expresión genética, observamos que la mayoría de trabajos se centran en realizar una severa selección de genes para posteriormente aplicar el clasificador con el que se trabaja. El motivo es que aquí no se busca construir la red genética para encontrar una explicación sino que lo que se busca es aumentar la precisión del clasificador pues, no en vano, en la gran mayoría de los trabajos se realiza la clasificación sobre muestras de expresión genética de distintos tipos de cáncer.

No obstante, si consideramos que no sólo sería interesante centrarse en la precisión del pronóstico por parte del clasificador, también sería muy conveniente que los clasificadores (como hace una red genética) nos explicasen qué mecanismos de regulación genética se están produciendo en aquellas muestras etiquetadas como cancerígenas.

El clasificador naïve bayes ha sido utilizado por algunos autores para evaluar distintos métodos de selección de características o, también, como método de referencia para comparar con otros clasificadores. Por ejemplo, Inza y col. [175] utilizan cuatro métodos de aprendizaje supervisado (IB1, naïve bayes, C4.5 y CN2). Su estudio se centra en la selección de genes

que mejores resultados nos dan en los clasificadores. Para ello utilizan una técnica de envoltura en cada uno: realizan una búsqueda secuencial hacia delante. Para evaluar la precisión de los clasificadores utilizan validación cruzada dejar-uno-fuera, aplicando su estudio a los conjuntos de datos sobre leucemia [136][13], cáncer de colon [12][14] y sobre diferentes tipos de cáncer y medicamentos [293].

Una extensión al trabajo anterior se realiza en [173] donde aplican varias técnicas de filtrado y de envoltura. Las técnicas de filtrado que utilizan son continuas (métrica-p y métrica-t) y discretas (entropía de Shannon, distancia euclídea, dependencia de Kolmogorov y divergencia de Kullback-Leibler). En este caso se discretizan los datos en tres estados. Para validar los resultados utilizan validación cruzada dejar-uno-fuera con los clasificadores IB1, naïve bayes, C4.5 y CN2. En el enfoque de envoltura utilizan búsqueda secuencial hacia delante tanto para los datos continuos como para los discretizados. En las técnicas de filtrado se quedan con subconjuntos de 3, 5, 10 y 20 genes (los más relevantes). Obtienen mejores resultados con las técnicas de envoltura y cuando discretizan los datos.

En cambio, en [44] también aplican una selección de genes a las bases de datos de [12] y de [136] pero utilizando EDAs [208] con cuatro tipos distintos de inicialización. Para la búsqueda utilizan un naïve bayes selectivo.

También con el objetivo de evaluar distintos métodos de selección de genes, encontramos el trabajo de Ben-Dor, Fiedman y Yakhini [33] donde se presentan diferentes métodos (métrica TNoM, información mutua, pérdida logarítmica, distribución y valores p, predicción logística, métrica de separación gaussiana) para seleccionar los genes más relevantes. Para evaluar cada método utilizan los genes seleccionados en la construcción de naïve bayes. Utilizan la precisión del clasificador obtenida mediante validación cruzada

[13]El conjunto sobre leucemia presentado en [136] es el más usado en clasificación supervisada de datos de expresión genética. En este conjunto de datos tenemos 72 muestras de pacientes con leucemia, incluyendo en el estudio 7129 genes. Las distintas muestras están clasificadas según el tipo de leucemia que sufre el paciente: 25 pacientes con leucemia mieloide aguda (AML, por su siglas en inglés: *acute myelogenous leukemia*) y 47 pacientes con leucemia linfoide aguda (ALL: por su siglas en inglés *acute lymphoblastic leukemia*). Este conjunto de datos junto con otros está disponible en el programa contra el cáncer del Instituto Tecnológico de Massachusetts (MIT) en http://www.broad.mit.edu/MPR, concretamente en http://www.broad.mit.edu/mpr/data_set_ALL_AML.html

[14]Disponible en http://microarray.princeton.edu/oncology/affydata/index.html (62 casos, 2000 genes, 2 clases)

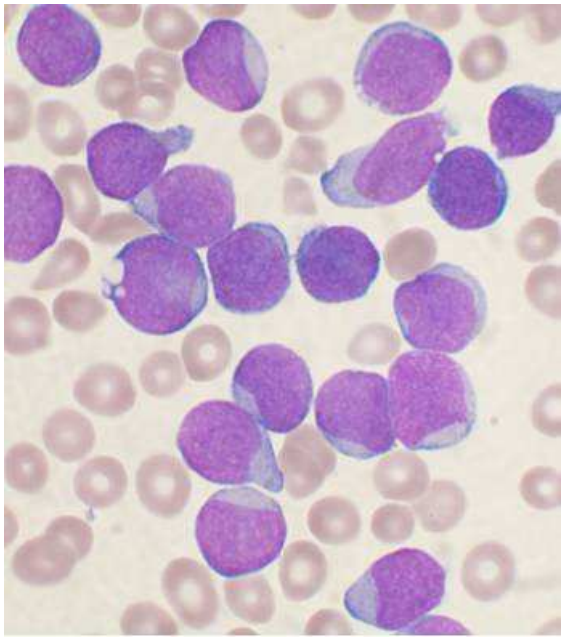

Figura 2.15: Muestra de la médula de un paciente con leucemia linfoide aguda (Fuente Wikimedia http://commons.wikimedia.org/wiki/File:Acute_leukemia-ALL.jpg).

dejar-uno-fuera como medida para evaluar los métodos de selección usados. Los aplican a los datos de cáncer de colon [12] y leucemia [136].

En [34], los mismos autores utilizan de forma similar el naïve bayes pero en lugar de selección de genes lo hacen para validar las clases encontradas por métodos no supervisados en las bases de datos de leucemia [136] y linfoma [9][15].

En [237] Marmor y col. utilizan varios clasificadores, entre ellos el naïve bayes con caracter puramente comparativo, para varios datos de cáncer (leucemia [136], linfoma difuso de células B grandes [314][16], cáncer de próstata

[15] El conjunto de Alizadeh y col. [9] sobre el linfoma difuso de células B grandes, es otra de las bases de datos más utilizadas. Se puede encontrar en http://llmpp.nih.gov/lymphoma/ (46 casos clasificados, 4096 genes, 2 clases)

[16] Esta clase de linfoma es el cáncer maligno del sistema linfático que más se da en adultos, curable sólo en menos del 50 % de los casos. Este conjunto de datos se puede encontrar también en el programa contra el cáncer del MIT, disponible en http://www.broad.mit.edu/mpr/lymphoma/ (77 muestras, 7070 genes, 2 clases )

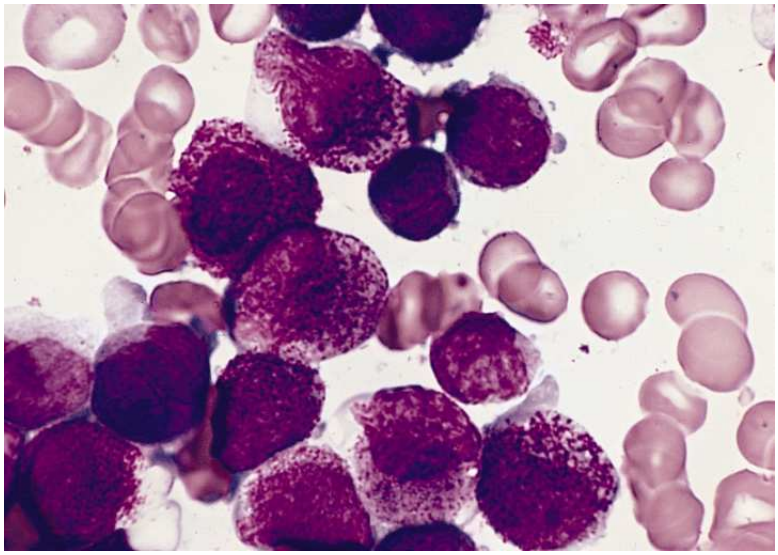

Figura 2.16: Muestra de la médula de un paciente con leucemia mieloide aguda (Fuente Wikimedia http://commons.wikimedia.org/wiki/File:AML-M3.jpg).

[316][17], leucemia de linaje mixto [18][18], tumores de célula pequeña y redonda de color azul [183][19] y cáncer de pulmón [38][20]). Con su método obtienen mejores resultados que el resto de autores con los que se compara.

En [254] Osareh y Shadgar también comparan distintos clasificadores, entre ellos naïve bayes y distintos métodos de selección de genes de tipo filtrado (información mutua condicionada, creencia, análisis de componentes principales, estadístico t). Para evaluar sus propuestas usan bases de datos de cáncer (cáncer de pulmón, cáncer de próstata, leucemia, cáncer de ma-

[17]Datos disponibles también en el MIT en http://www.broad.mit.edu/publications/broad895 (102 muestras, 12533 genes, 2 clases )

[18]Los datos de la leucemia de linaje mixto (o MLL del inglés, *mixed lineage leukemia*), se pueden encontrar en http://www.broad.mit.edu/publications/broad901 (72 muestras, 12533 genes, 3 clases)

[19]El conjunto de datos de tumores de célula pequeña y redonda de color azul (SRBCT, del inglés *small round blue cell tumors*) se puede encontrar en http://home.ccr.cancer.gov/oncology/oncogenomics/ (83 casos, 2308 genes, 4 clases)

[20]Disponible en http://www.broad.mit.edu/publications/broad903 (203 muestras, 12600 genes, 5 clases)

ma [151][21], leucemia de linaje mixto, tumores de célula pequeña y redonda de color azul, tumores cerebrales [275][22], cáncer de colon, cáncer de ovario [308][23] y un conjunto de datos con 14 tumores distintos [281][24]).

La simplicidad y rapidez del clasificador naïve bayes ha propiciado su utilización en este tipo de conjunto de datos tan problemáticos y costosos computacionalmente de utilizar. Ése es el caso de Moler y col. en [236] donde utilizan naïve bayes (supervisado y no supervisado) y SVM en el estudio de datos expresión genética provenientes de muestras del cáncer de colon [12] . Los autores primero utilizan un naïve bayes no supervisado usando gaussianas para detectar subclases dentro de los datos. Para fijar el número de clases utilizan el método propuesto en [67]. Obtienen dos medidas *Naïve Bayes Relevance* -NBR- (obtenida a partir del naïve bayes no supervisado construido) y *Naïve Bayes Global Relevance* -NBGR- (obtenido a partir de NBR), y a partir de los cuales se ordenan los genes. La medida NBR nos da el grado en que un gen distingue entre dos clases, mientras que la medida NBGR nos calcula el grado en el que un gen distingue todas las clases. Con la medida NBGR construyen distintos conjuntos de genes, a los cuales les aplican una validación cruzada dejar-uno-fuera usando SVM como clasificador. Los mejores resultados se obtienen con un conjunto de los 200 mejores genes.

En [109], Fan y col. utilizan primero una selección de genes por filtrado (*stepwise regresion-based feature selection*), más una transformación de atributos (CC-ICA) y posteriormente construyen un clasificador naïve bayes. La transformación de atributos CC-ICA (*Class-conditional independente componente analysis*), transforma los atributos en nuevas variables que son independientes condicionalmente dada la clase (una de las restricciones impuesta por el naïve bayes). Según el autor aunque muchos métodos de selección de genes se centran en la relación del gen con la variable clase, métodos basados en la redundancia mínima y máxima relevancia pueden seleccionar genes más

[21]Disponible en http://research.nhgri.nih.gov/microarray/NEJM_Supplement/ (22 casos, 3226 genes, 2 clase)

[22]Disponible en http://www.gems-system.org/ (90 casos, 5920 genes, 5 clases)

[23]Disponible en http://www.c2i.ntu.edu.sg/Resources/Download/Software/Zhu_Zexuan/Datasets.html (253 casos, 14154 genes, 2 clases)

[24]Disponible en http://www.broad.mit.edu/publications/broad900 (308 casos, 15009 genes, 26 clases )

representativos [97, 259], por eso utilizan *stepwise regresion-based feature selection*. Aplican su método a los conjuntos de cáncer de pulmón, de colon y leucemia.

Sin ser un naïve bayes pero con un esquema estructural similar, en [152] Helman y col. hacen una búsqueda dentro del manto de Markov de la variable a clasificar, restringida al conjunto de nodos padre de la clase, es decir, restringen el problema del aprendizaje estructural a encontrar el subconjunto de mejores padres de la clase, basándose en la idea de que sólo unos pocos genes sirven para clasificar. Esta búsqueda la realizan de dos formas: en la primera introducen una variación de la búsqueda secuencial hacia delante, usando una métrica bayesiana para evaluar cada uno de los posibles nodos a añadir. En la segunda, utilizan una selección de genes externa a la red bayesiana para obtener un pequeño conjunto de genes. Para ello calculan un valor de calidad de separación (similar a la métrica TNoM de [32, 34]) para cada gen y los ordenan usando dicho valor, quedándose sólo con los más relevantes para construir la red bayesiana. En los experimentos utilizan los datos de cáncer de leucemia del MIT [136] y los datos de cáncer de colon de [12] discretizados a tres valores.

También intentando tratar con la naturaleza continua de los datos de expresión genética, en [61] Cano y col. utilizan un naïve bayes de gaussianas, que aplican al estudio del cáncer de linfoma [363]. Realizan una selección de genes en dos pasos: primero aplican una selección por filtrado basada en la función ANOVA para el análisis de la varianza. De esta forma se obtienen los genes más relevantes no correlados. En el segundo paso, a partir del subconjunto encontrado, realizan una selección de genes usando un naïve bayes (enfoque de envoltura) en diferentes particiones, obteniendo en cada partición un subconjunto de genes. Posteriormente, en la denominada fase de abducción, se construye una red bayesiana, usando el algoritmo K2, que codifica la distribución de probabilidad conjunta sobre los distintos conjuntos de genes encontrados en la fase de envoltura. El objetivo de la fase de abducción es incrementar la robustez del proceso, ya que con la red aprendida se calculan las configuraciones de genes más probables. Concluyen que utilizando un clasificador simple (naïve bayes) sobre el conjunto final de genes se obtienen muy buenos resultados en clasificación supervisada, con la consiguiente reducción del número de genes a estudiar.

Mientras que en [62], los mismos autores también utilizan naïve bayes de

envoltura y gaussianas, pero ahora con una fase de preordenamiento de genes y otra de eliminación de variables irrelevantes. En la fase de preordenamiento se realiza un ranking de los genes usando tres métodos: preordenamiento aleatorio, preordenamiento ANOVA (se ordenan en función de su coeficiente ANOVA), preordenamiento por precisión (para cada variable se construye un clasificador con esa sola variable y se calcula su precisión). Esta ordenación mejora la clasificación del naïve bayes de envoltura y además les permite desechar aquellas variables con peor valor en el ranking. En la fase de eliminación de variables irrelevantes definen un método heurístico para detectar variables irrelevantes. Entonces eliminan aquellas variables que puedan ser irrelevantes respecto a las ya obtenidas. Aplican su estudio al cáncer de linfoma utilizando los conjuntos de [363] y [9]. En este último se realiza una selección de genes previa, como la descrita en [61]. Concluyen que con la metodología propuesta se han realizado importantes mejoras, reduciendo el coste computacional y minimizando la cantidad de genes seleccionados.

Como vimos había distintos clasificadores que mejoraban el clasificador naïve bayes. Uno de estos clasificadores era el clasificador bayesiano k-dependiente que tenía el objetivo de superar la restricción de que los atributos son independientes entre sí dada la clase. En [17] Armañanzas y col. utilizan un clasificador KDB, con $k = 4$. Realizan bootstrap (1000 iteraciones) para dar robustez al proceso y en cada iteración hacen una selección de características (*correlation feature selection* [141]) seguida de la construcción del clasificador. Al final del proceso, el clasificador se construye promediando los diferentes clasificadores obtenidos y considerando los enlaces que aparezcan un número determinado de veces para dar robustez a los resultados. En [14] se centran en la selección de genes, realizada con *selección de genes por consenso* [16], pero también utilizando un clasificador k-dependiente y bootstrap.

Según la exposición de los distintos clasificadores que vimos, el siguiente clasificador en términos de capacidad expresiva sería el clasificador BAN, el cual es utilizado por Bosin y col. en [48]. Se usa la métrica MDL para hacer un ranking de genes. Posteriormente a partir de dicha ordenación se realiza una selección de genes. Construyen dos clasificadores, un naïve bayes y un BAN con los genes más importantes según el ranking hecho e iterativamente van añadiendo más genes a cada modelo (siguiendo el orden) hasta que no mejora.

Además de los clasificadores bayesianos donde todas las variables son

nodos hijos de la clase (como el naïve bayes, el clasificador KDB o el BAN), también se han utilizado redes bayesianas sin ningún tipo de restricción estructural para realizar clasificación supervisada. Por ejemplo en [163], Hwang, Zhang y col. comparan la capacidad predictora de árboles neuronales, redes de base radial (RBF) y redes bayesianas, en la clasificación de distintos tipos de cáncer de leucemia [136]. En el caso de las redes bayesianas, discretizan los datos en dos valores y también se realiza una selección de genes: se van a quedar con tan sólo cuatro; para escoger dos se usa la información mutua (los cuatro no es posible porque todos aparecen sobreexpresados para una sola clase) y para escoger los otros dos genes se usa la métrica p [318] (sobreexpresados para la otra clase). Para construir la red bayesiana utiliza un algoritmo métrica+búsqueda, encontrando la mejor red (como son sólo cuatro variables más la clase, se puede hacer una búsqueda exhaustiva) aunque poniendo como restricción que cada nodo sólo puede tener menos de tres padres.

Siguiendo la misma metodología, en [367] para la selección de genes sólo utilizan métrica-p, obteniendo 50 genes de conjuntos de datos sobre cáncer de leucemia [136] y sobre cáncer de colon [12], discretizados a dos valores. Comparan las redes bayesianas con distintos clasificadores ( C4.5, perceptrones multicapa, Support Vector Machines y Weighted voting) obteniendo los mejores resultados en tanto por ciento de bien clasificados con las redes bayesianas.

Los mismos autores, usan redes bayesianas sin ningún tipo de restricción estructural en [65] pero, ahora, conjuntamente con STVQ (*Soft Topographic Vector Quitization*) [138] (una técnica de agrupamiento), aplicándolo al conjunto de datos NCI60 [301] (datos de expresión genética y medicamentos, clasificados en nueve tipos distintos de cáncer). Discretizan los datos en tres valores. Para los experimentos utilizan un conjunto de datos recortado y dos versiones aún más reducidas: una, reducción con prototipos, donde se agrupan los genes y medicamentos por separado, utilizándose el atributo que está en el centro de cada agrupamiento para el nuevo conjunto (obtiene 40 genes y 5 compuestos); y dos, reducción con atributos seleccionados, donde se agrupan genes y medicamentos de forma conjunta y se seleccionan todos los miembros de algunos agrupamientos adyacentes (se quedan con 12 genes y cuatro compuestos). Los agrupamientos los han creado mediante STVQ. Para la clasificación de los datos utilizan dos algoritmos de aprendizaje: una búsqueda local con métricas que sólo aplican a los dos conjuntos reducidos, y una búsqueda heurística. En la búsqueda heurística se realiza una búsqueda

local para cada nodo en su manto de Markov, limitando el tamaño del mismo. Concluyen que los resultados obtenidos coinciden con hechos biológicos demostrados.

Bockhorst y col. [47] utilizan una red bayesiana para predecir operones en el genoma de la *E. Coli*, utilizando los datos de [45]. En la identificación de operones utilizan diversas fuentes de datos, entre ellos, microarrays de ADN. En un anterior trabajo [82] utilizan un naïve bayes para esta tarea. En este caso utilizan una red bayesiana sin restricciones cuya estructura está hecha a mano y los parámetros de la red se han aprendido usando un algoritmo de aprendizaje paramétrico. Comparan ambas redes (naïve bayes y red bayesiana sin restricciones generada a mano) y ven que la red bayesiana obtiene mejores resultados y además esta diferencia es significativa estadísticamente.

Diao y col. presentan un sistema (*Disease Gene Explorer*), en [95], basado en tres pasos: agrupamiento, aprendizaje de redes bayesianas y selección de genes de una enfermedad. Realizan el paso de agrupamiento con la idea de evitar sobreajuste a los datos, utilizan un algoritmo basado en las k-medias y en el test-t. Usan dos tipos de aprendizaje de redes bayesianas, una para los genes dentro del agrupamiento y otra con genes a lo largo de diferentes agrupamientos.

Como vimos, la utilización de información adicional puede ayudar en la construcción de las redes bayesianas cuando trabajamos con conjuntos de datos tan escasos y con tantas variables. Si lo que aprendemos es una red bayesiana que va a trabajar como clasificador, podemos hacer justamente lo mismo, como ocurre en el trabajo presentado por Gevaert y col. [131], donde para cada microarray hay información del historial médico, el cual es usado de tres formas para construir el clasificador: una, construye el clasificador sólo con datos de expresión genética; dos, construye dos clasificadores uno con los datos de expresión genética y otro con los datos del historial médico, usando ambos clasificadores para hacer predicciones; tres, construye dos clasificadores por separado y después del aprendizaje estructural, los une por la variable clase. Para construir las redes usa la métrica BDe y la búsqueda voraz del algoritmo K2. En [132] usan como información previa datos obtenidos de Pubmed[25] pero en este caso dicha información previa la incorporan

[25]Es un archivo digital sobre artículos de investigación biomédica y de ciencias de la vida. Disponible en http://www.ncbi.nlm.nih.gov/pubmed/

en la distribución a priori del aprendizaje estructural.

En [164] Hwang y col. utilizan también redes bayesianas sin restricciones estructurales como clasificadores, pero para evitar el sobreaprendizaje, obtienen una red promediada a partir de distintos ordenamientos de las variables (usando *Bayesian Model Averaging*) . Según los autores este método es costoso en tiempo pero preciso en conjuntos con pocos datos y mucho ruido.

Intentando representar información temporal, construyendo un híbrido entre clasificadores bayesianos y redes bayesianas dinámicas tenemos el trabajo de Tucker y col. [342], donde utilizan naïve bayes, un clasificador BAN y tres clasificadores bayesianos para usar con datos temporales: en el primero se tiene una variable temporal que es el padre de todos los atributos, en el segundo los de intervalos anteriores pueden estar relacionados con genes del intervalo actual (como en las redes bayesianas dinámicas) mediante arcos temporales, en el tercer tipo de clasificador se usan los dos anteriores enfoques de forma combinada. Para aprender una red BAN se usa enfriamiento simulado y la métrica MDL. Se obtiene que los clasificadores mejoran un poco con el uso de información temporal.

## 2.6. Discusión.

La técnica de microarrays de ADN ha permitido medir de forma simultánea el grado de activación de los genes (miles) de un organismo. Las redes bayesianas se han aplicado con éxito al tratamiento de este tipo de datos, siendo la mejor herramienta que hay en la literatura para la inferencia en redes de regulación genética.

El éxito de la aplicación de las redes bayesianas a los datos de expresión genética se debe a una serie de ventajas: poseen sólidas bases estadísticas, determinan relaciones estocásticas entre genes, pueden describir procesos locales, proporcionan modelos de influencia causal, permiten la incorporación de conocimiento experto, se pueden usar en clasificación, son modelos visualmente interpretables, utilizan la probabilidad como medida de incertidumbre por lo que son indicadas para trabajar con datos con gran cantidad de ruido, entre otras ventajas ya comentadas.

Aún así los datos de expresión genética presentan una serie de graves problemas que los distintos autores han intentado sortear de una forma u otra. El principal problema es la dimensionalidad de los datos (pocos casos, muchas variables). Este obstáculo es resuelto por algunos autores utilizando diversas técnicas no excluyentes entre sí:

- Algoritmos de aprendizaje adaptados a un gran número de variables. Normalmente se basan en la idea de limitar el número de padres o vecinos por nodo, lo cual no supone una pérdida en la calidad de las estructuras aprendidas, ya que las redes genéticas están escasamente conectadas. Además la mayoría de los autores se decantan por un enfoque métrica+búsqueda.
- Utilizar un método de bootstrapping (o algún tipo de promediado de diversas redes obtenidas) que de robustez a la red final, evite el sobreajuste a los datos y permita distinguir relaciones correctas del ruido.
- Uso de conocimiento experto o conocimiento adicional que nos permita suplir la escasez de datos, mejorar el resultado y reducir el coste computacional.
- Organización modular de los genes para reducir el número de variables a tratar (clúster o módulos en lugar de genes) y además agrupar genes con funciones biológicas similares o relacionadas.
- Selección de características. Casi todos los autores han reducido el número de genes a estudiar aunque sea, al menos, quedándose con una proporción alta de genes.

Otro de los problemas que vimos que presentaban los datos provenientes de microarrays de ADN era el hecho de que los niveles de expresión se movían en un rango de valores continuos. La mayoría de los autores optan por una discretización en tres estados[26]. La discretización tiene la ventaja de que es

[26]Téngase en cuenta que el modelo de red de regulación genética que se utiliza hoy en día está basado en el modelo de Kauffman [180] donde los genes sólo tienen dos estados, encendido o apagado. A pesar de ser un modelo simplificado ha resultado ser bastante útil desde el punto de vista predictivo, permitiendo deducir resultados biológicamente relevantes.

más estable frente a las variaciones aleatorias o sistemáticas que se producen en los niveles de expresión de los microarrays. Pero tiene el inconveniente de que puede producirse una pérdida de información. Por ello, algunos autores han propuesto la utilización de modelos basados en gaussianas, mixturas de gaussianas o regresión no paramétrica.

Las redes biológicas tienen ciclos y bucles de retroalimentación, ciclos que las redes bayesianas no puede representar. No obstante, las redes bayesianas dinámicas nos van a permitir representar auto regulación y, además, regulación en el tiempo.

Dentro del uso de clasificadores bayesianos para el análisis de datos de expresión genética, observamos una clara tendencia a utilizar redes bayesianas generales como clasificadores. Con generales nos referimos a que no tienen restricciones estructurales como, por ejemplo, el clasificador naïve bayes o el clasificador BAN. La idea que soporta esta tendencia es que no sólo es importante obtener un clasificador con una gran precisión predictora, además es conveniente que el clasificador nos permita explicar los mecanismos de regulación genética que se están produciendo.

# Capítulo 3

# Incorporación de conocimiento experto.

Hay un gran número de trabajos sobre el aprendizaje automático de redes bayesianas a partir de datos, pero se ha prestado poca atención al uso de conocimiento experto en dicho proceso. Con conocimiento experto queremos referirnos a aquel conocimiento que no esté en los datos y que, normalmente, posee un experto sobre el dominio del problema que estemos tratando. Dicho conocimiento experto, si lo utilizamos en conjunción con un algoritmo de aprendizaje automático puede ayudar en el proceso de obtención de la red y contribuir a obtener resultados más precisos e incluso reducir el espacio de búsqueda, esto es, mejores resultados en menos tiempo.

En los últimos años y especialmente en el análisis de datos de expresión genética, se ha intentando incorporar conocimiento adicional que ayude en el aprendizaje de las redes bayesianas. No obstante, cada autor ha incorporado dicho conocimiento probando distintas aproximaciones. Pero no existe una forma sistemática o estándar de hacerlo, si bien, muchos autores proponen usar dicha información en la distribución a priori de la métrica usada en el algoritmo de aprendizaje, este método es bastante poco intuitivo desde el punto de vista del experto. Es más, incorporar el conocimiento en forma de probabilidades es un proceso bastante subjetivo, como ya observamos al hablar de redes bayesianas elicitadas por el experto.

Las redes bayesianas son visualmente interpretables; basándonos en esta ventaja, nuestro objetivo será incorporar conocimiento adicional de forma

gráfica (y en consonancia con la interpretabilidad del modelo) haciendo que el proceso sea intuitivo y, por tanto, simplificando la extracción y utilización de conocimiento experto. Y ésto se va a hacer en forma de restricciones estructurales.

En este capítulo se va estudiar el uso de restricciones para algoritmos de aprendizaje de la estructura de una red bayesiana. Estas restricciones podrán codificar el conocimiento de un experto para un dominio dado, de forma que una red bayesiana representando este dominio deba satisfacer dichas restricciones. En concreto, vamos a definir y considerar tres tipos de restricciones: (1) existencia de arcos y aristas, (2) ausencia de arcos y aristas, y (3) restricciones de orden. Todas ellas serán consideradas restricciones fuertes (en oposición a las restricciones suaves de [150]), en el sentido de que se asumen ciertas para la red bayesiana representando el dominio de conocimiento, y por lo tanto todas las redes bayesianas candidatas deben cumplirlas.

## 3.1. Notación y preliminares.

Consideremos un conjunto finito $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$ de variables discretas y aleatorias. Como vimos, la estructura de una red bayesiana es un grafo dirigido acíclico (DAG, del inglés *Directed Acyclic Graph*) $G = (\mathbf{X}, E_G)$, donde $\mathbf{X}$ es el conjunto de nodos[1] y $E_G$ representa el conjunto de arcos en el grafo.

El aprendizaje de la estructura de una red bayesiana consiste en, dado un conjunto de datos $\mathcal{D}$ con instancias de las variables de $\mathbf{X}$, encontrar la red que, en algún sentido, mejor represente a $\mathcal{D}$. Como vimos, los algoritmos de aprendizaje estructural de una red bayesiana se podían dividir en dos metodologías: basados en métrica+búsqueda y basados en tests de independencia.

Los algoritmos basados en métrica+búsqueda, utilizaban distintas métricas y distintos algoritmos de búsqueda, pero estos, normalmente tienen un mismo espacio de búsqueda: el espacio de búsqueda de los DAGs. Aunque hay otras alternativas, como la búsqueda en el espacio de clases equivalentes de DAGs [6, 72] o el espacio de órdenes [90, 207], nos vamos a centrar en el

[1] No se va a distinguir entre un nodo y la variable que representa.

espacio de búsqueda de los DAGs.

Por otro lado, los métodos basados en tests de independencia buscan la mejor solución dentro del espacio de los grafos acíclicos parcialmente dirigidos. Donde, un *grafo parcialmente dirigido y acíclico* (PDAG, del inglés *Partially Directed Acyclic Graph*) es un grafo que puede contener enlaces dirigidos (arcos) o enlaces no dirigidos (aristas), pero no puede contener un ciclo formado por enlaces dirigidos.

## 3.2. Tipos de restricciones.

Vamos a estudiar tres tipos de restricciones en las estructuras de la red que denominaremos restricciones de existencia, de ausencia y de orden.

### 3.2.1. Restricciones de existencia.

Vamos a considerar dos tipos de restricciones de existencia, la existencia de arcos y la existencia de aristas. Sean $\mathcal{E}_a, \mathcal{E}_e \subseteq \mathbf{X} \times \mathbf{X}$ dos subconjuntos de pares de variables, con $\mathcal{E}_a \cap \mathcal{E}_e = \emptyset$. Se interpretan como sigue:

- $(X, Y) \in \mathcal{E}_a$: el arco $X \rightarrow Y$ debe pertenecer a cualquier grafo del espacio de búsqueda.
- $(X, Y) \in \mathcal{E}_e$: existe un enlace dirigido o no entre dos nodos $X$ e $Y$ en cualquier grafo del espacio de búsqueda. En el caso del espacio de búsqueda de un DAG, significa que el arco $X \rightarrow Y$, o bien, el arco $Y \rightarrow X$ debe aparecer en cualquier DAG.

Un ejemplo del uso de restricciones de existencia puede ser un algoritmo para construir el clasificador BAN [69], donde se fije la estructura naïve bayes (arcos desde la clase a todos los atributos) y se realice una búsqueda para añadir arcos que unan sólo a los atributos.

### 3.2.2. Restricciones de ausencia.

Vamos a considerar dos tipos de restricciones de ausencia: ausencia de arcos y ausencia de aristas. Sean $\mathcal{A}_a, \mathcal{A}_e \subseteq \mathbf{X} \times \mathbf{X}$ dos subconjuntos de pares de variables, con $\mathcal{A}_a \cap \mathcal{A}_e = \emptyset$. Su significado es el siguiente:

- $(X, Y) \in \mathcal{A}_a$: el arco $X \rightarrow Y$ no puede estar presente en ningún grafo del espacio de búsqueda.
- $(X, Y) \in \mathcal{A}_e$: no existe un enlace dirigido o no dirigido uniendo los nodos $X$ e $Y$ en ningún grafo del espacio de búsqueda. En el espacio de búsqueda de los DAG esto significa que no pueden estar presentes en ningún DAG, ni el arco $X \rightarrow Y$ ni el arco $Y \rightarrow X$ .

Un ejemplo del uso de restricciones de ausencia puede ser el clasificador naïve bayes selectivo [205], donde se prohiben aristas entre los atributos y arcos de los atributos hacia la clase.

### 3.2.3. Restricciones de orden.

Necesitamos conceptos adicionales para una mejor compresión de este tipo de restricción. Podemos decir que un orden total, $\sigma$, del conjunto de variables $\mathbf{X}$ es *compatible* con un orden parcial, $\mu$, del mismo conjunto si:

$$\forall X, Y \in \mathbf{X}, \text{ si } X <_\mu Y \text{ entonces } X <_\sigma Y$$

Es decir, si $X$ precede a $Y$ en el orden parcial $\mu$ entonces también $X$ precede a $Y$ en el orden total. Obsérvese que un DAG (y también un PDAG) $G$ determina un orden parcial en sus variables: si hay un camino dirigido de $X$ a $Y$ en $G$, entonces $X$ precede a $Y$. Por lo tanto, también podemos decir que un orden total $\sigma$ en el conjunto $\mathbf{X}$ es *compatible* con un grafo $G = (\mathbf{X}, E_G)$ si

$$\forall X, Y \in \mathbf{X}, \text{ si } X \rightarrow Y \in E_G \text{ entonces } X <_\sigma Y$$

Un subconjunto $\mathcal{R}_o$, $\mathcal{R}_o \subseteq \mathbf{X} \times \mathbf{X}$ define una restricción de orden, cuya interpretación es:

- $(X, Y) \in \mathcal{R}_o$: todo grafo del espacio de búsqueda tiene que satisfacer que $X$ precede a $Y$ en algún orden total de las variables compatible con el grafo.

Obsérvese que la restricción es equivalente a afirmar que no hay un camino dirigido de $Y$ a $X$ en ningún grafo del espacio de búsqueda. Las restricciones de orden pueden representar, por ejemplo, precedencia temporal o funcional entre variables. Ejemplos de uso de restricciones de orden son todos aquellos algoritmos de aprendizaje de redes bayesianas que requieren un orden total de las variables (como el algoritmo K2 [80] o el algoritmo que aparece en [89]).

## 3.3. Representación de restricciones.

Para trabajar con restricciones nos va a ser útil representarlas de forma gráfica. De este modo, las restricciones de existencia pueden ser representadas mediante un grafo parcialmente dirigido $G_e = (\mathbf{X}, E_e)$, donde cada elemento $(X, Y)$ de $\mathcal{E}_a$ se asocia con el correspondiente arco $X \rightarrow Y \in E_e$, y cada elemento $(X, Y)$ de $\mathcal{E}_e$ se asocia con la arista $X$—$Y \in E_e$. La figura 3.2 (a) muestra el grafo de restricciones de existencia usado por un clasificador BAN.

Las restricciones de ausencia se pueden representar mediante otro grafo parcialmente dirigido $G_a = (\mathbf{X}, E_a)$, donde los elementos $(X, Y)$ de $\mathcal{A}_a$ se corresponden con los arcos $X \rightarrow Y \in E_a$ y los elementos $(X, Y)$ de $\mathcal{A}_e$ se asocian con las aristas $X$—$Y \in E_a$. En la figura 3.2(b) se representa el grafo de restricciones de ausencia correspondiente a un clasificador naïve bayes selectivo.

Finalmente, las restricciones de orden se van a presentar usando un grafo dirigido $G_o = (\mathbf{X}, E_o)$, en el cual cada $(X, Y)$ de $\mathcal{R}_o$ se asocia con el arco $X \rightarrow Y \in E_o$. Nótese que, al asumir que las restricciones de orden forman un orden parcial (esto es, las relaciones son transitivas), no estamos forzados a incluir en $G_o$ un arco para cada elemento de $\mathcal{R}_o$. $G_o$ puede ser cualquier grafo que en su clausura transitiva contiene un arco para cada elemento de $\mathcal{R}_o$. Por ejemplo, para representar un orden total mediante restricciones $X_1 < X_2 < \ldots < X_n$ es suficiente incluir en $G_o$ los $n - 1$ arcos $X_i \rightarrow X_{i+1}$, $i = 1, \ldots, n - 1$, en lugar de tener un grafo completo con todos los arcos $X_i \rightarrow X_j$, $\forall i < j$. La figura 3.2 (c) muestra el grafo de restricciones de orden usado por el algoritmo K2.

En la implementación de la herramienta Elvira, la introducción de las restricciones se pude hacer desde el interfaz en forma de enlaces dirigidos o no como se puede apreciar en al figura 3.1.

## 3.4. Verificación de restricciones usando operaciones sobre enlaces.

Ahora, vamos a definir formalmente cuando un grafo dado es *consistente* con un conjunto de restricciones, es decir, cuando el grafo verifica las restric-

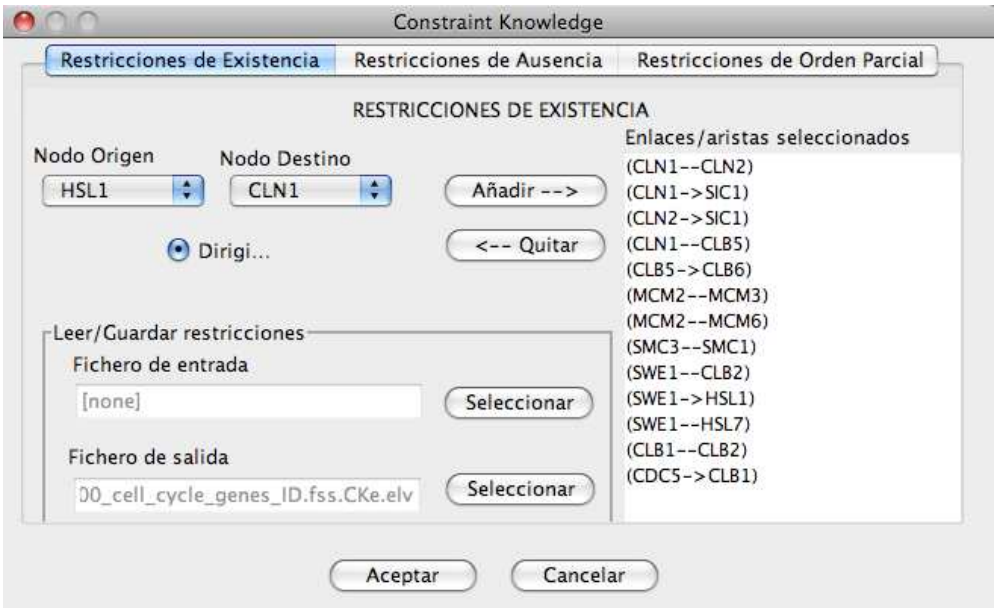


Figura 3.1: Introducción de restricciones desde la interfaz gráfica de Elvira.

ciones.

**Definición.** Sean $G_e = (\mathbf{X}, E_e)$, $G_a = (\mathbf{X}, E_a)$ y $G_o = (\mathbf{X}, E_o)$ los grafos que representan las restricciones de existencia, ausencia y orden, respectivamente. Sea $G = (\mathbf{X}, E_G)$ un DAG y $H = (\mathbf{X}, E_H)$ un PDAG. Decimos que:

1. $G$ es consistente con las restricciones de existencia si y sólo si
   - $\forall X, Y \in \mathbf{X}$, si $X \rightarrow Y \in E_e$ entonces $X \rightarrow Y \in E_G$,
   - $\forall X, Y \in \mathbf{X}$, si $X — Y \in E_e$ entonces $X \rightarrow Y \in E_G$ o $Y \rightarrow X \in E_G$.
2. $G$ es consistente con las restricciones de ausencia si y sólo y si
   - $\forall X, Y \in \mathbf{X}$, si $X \rightarrow Y \in E_a$ entonces $X \rightarrow Y \notin E_G$,
   - $\forall X, Y \in \mathbf{X}$, si $X — Y \in E_a$ entonces $X \rightarrow Y \notin E_G$ y $Y \rightarrow X \notin E_G$.
3. $G$ es consistente con las restricciones de orden si y sólo si
   - existe un orden total $\sigma$ de las variables en $\mathbf{X}$ compatible con los dos grafos $G$ y $G_o$.

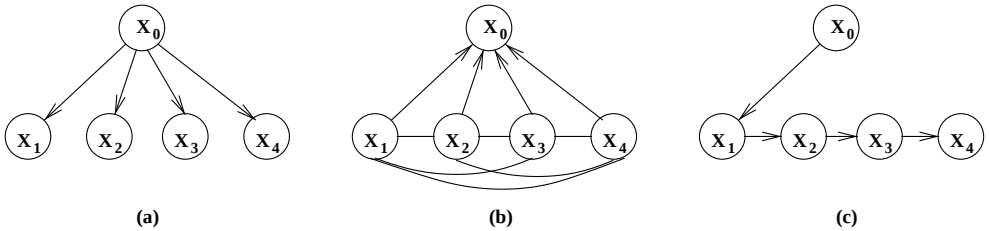


Figura 3.2: (a) Grafo de restricciones de existencia $G_e$ usado por un método de aprendizaje para construir un clasificador BAN; (b) Grafo de restricciones de ausencia $G_a$ correspondiente a un naïve bayes selectivo; (c) Grafo de restricciones de orden $G_o$ usado por el algoritmo K2. En los casos (a) y (b) $X_0$ representa la variable a clasificar.

4. $H$ es consistente con las restricciones de existencia si y sólo si

   - $\forall X, Y \in \mathbf{X}$, si $X \rightarrow Y \in E_e$ entonces $X \rightarrow Y \in E_H$,
   - $\forall X, Y \in \mathbf{X}$, si $X — Y \in E_e$ entonces $X \rightarrow Y \in E_H$ o $Y \rightarrow X \in E_H$ o $X — Y \in E_H$,
   - $H$ puede transformarse en un DAG, que sea consistente con las restricciones de existencia, orientando sus aristas.

5. $H$ es consistente con las restricciones de ausencia si y sólo si

   - $\forall X, Y \in \mathbf{X}$, si $X \rightarrow Y \in E_a$ entonces $X \rightarrow Y \notin E_H$,
   - $\forall X, Y \in \mathbf{X}$, si $X — Y \in E_a$ entonces $X \rightarrow Y \notin E_H$, $Y \rightarrow X \notin E_H$ y $X — Y \notin E_H$,
   - $H$ puede transformarse en un DAG, que sea consistente con las restricciones de ausencia, orientando sus aristas.

6. $H$ es consistente con las restricciones de orden si y sólo si

   - existe un orden total $\sigma$ de las variables de $\mathbf{X}$ compatible con los dos grafos $H$ y $G_o$,
   - $H$ puede transformase en un DAG, que sea consistente con las restricciones de orden, dirigiendo sus aristas.

## 3.5. Autoconsistencia de restricciones.

Cuando estamos especificando el conjunto de restricciones a usar para un dominio dado, se hace necesario asegurarnos que dichas restricciones pueden ser satisfechas. En este sentido, decimos que un conjunto de restricciones es *autoconsistente* si hay algún DAG que es consistente con ellas. Comprobar la autoconsistencia de cada uno de los tres tipos de restricciones por separado es bastante simple:

**Proposición 3.5.1** *Sean $G_e = (\mathbf{X}, E_e)$, $G_a = (\mathbf{X}, E_a)$ y $G_o = (\mathbf{X}, E_o)$ los grafos que representan las restricciones de existencia, ausencia y orden, respectivamente. Entonces*

*(a) El conjunto de restricciones de existencia es autoconsistente si y sólo si el grafo $G_e$ no tiene ningún ciclo dirigido.*

*(b) El conjunto de restricciones de ausencia siempre es autoconsistente.*

*(c) El conjunto de restricciones de orden es autoconsistente si y sólo si $G_o$ es un DAG.*

**Demostración.** (a) *Condición necesaria*: sabemos que existe un DAG $G$ que es consistente con las restricciones. Si el grafo $G_e$ tiene algún ciclo dirigido, como todos los arcos de $G_e$ también deben ser arcos en $G$, entonces $G$ contendría un ciclo dirigido también, lo cual contradice el hecho de que $G$ sea un DAG.

*Condición suficiente*: sabemos que $G_e$ no tiene ningún ciclo dirigido. Empezando desde $G_e$ vamos a construir un grafo $G = (\mathbf{X}, E_G)$ como sigue: $\forall X, Y \in \mathbf{X}$, si $X \rightarrow Y \in E_e$ entonces $X \rightarrow Y \in E_G$; si $X \rightarrow Y \notin E_e$, $Y \rightarrow X \notin E_e$ y $X$—$Y \notin E_e$ entonces $X \rightarrow Y \notin E_G$ y $Y \rightarrow X \notin E_G$; si $X$—$Y \in E_e$ entonces incluimos en $E_G$ el arco $X \rightarrow Y$ o bien, el arco $Y \rightarrow X$. Este grafo $G$ es obviamente consistente con las restricciones. Siempre podemos seleccionar al menos una orientación de las aristas $X$—$Y$ que no genera ningún ciclo dirigido y por lo tanto $G$ será un DAG: si el arco $X \rightarrow Y$ genera un ciclo dirigido es debido a que había un camino dirigido de $Y$ a $X$ antes de incluir este enlace; si el arco $Y \rightarrow X$ también genera un ciclo dirigido, entonces también había un camino dirigido de $X$ a $Y$. Como tenemos un camino dirigido de $X$ a $Y$ y otro camino dirigido de $Y$ a $X$, entonces tendríamos un ciclo dirigido antes de introducir cualquier arco.

(b) El grafo vacío $G_\emptyset$ siempre verifica las restricciones de ausencia.

(c) *Condición necesaria*: sabemos que hay un DAG $G$ y un orden total compatible con ambos grafos $G$ y $G_o$. Si $G_o$ no es un DAG, entonces tiene un ciclo dirigido, y entonces no hay ningún orden total compatible con $G_o$.
*Condición suficiente*: sabemos que $G_o$ es un DAG. Hay al menos un orden total compatible con $G_o$. Por lo tanto para el grafo $G = G_o$ esta ordenación es obviamente compatible con $G$ y $G_o$. ∎

Cuando se consideran diferentes tipos de restricciones autoconsistentes simultáneamente, algunas interacciones pueden ocurrir entre los conjuntos de restricciones implicadas. Estas interacciones pueden originar inconsistencias. Por ejemplo, la existencia y ausencia de los mismos arcos; las restricciones de orden también pueden contradecir la existencia de algunos arcos (en tanto en cuanto implícitamente también representan restricciones de orden parcial). Por ejemplo, $X \to V$, $V \to Y \in E_e$ se contradice con $Y \to Z$, $Z \to T$, $T \to X \in E_o$.

Igualmente es posible que restricciones de ausencia o de orden fuercen una restricción de existencia. Por ejemplo, si un arco puede existir en una dirección u otra (es decir, tenemos $X—Y \in E_e$) pero una restricción de ausencia, o bien una restricción de orden, indica que una de las direcciones está prohibida (por ejemplo, $X \to Y \in E_a$ o $Y \to X \in E_o$), entonces la otra dirección es obligatoria ($X—Y$ debe ser reemplazada por $Y \to X$ en $E_e$).

De igual manera también se pueden producir interacciones entre los tres tipos de restricciones, dando lugar a inconsistencias. Por ejemplo, $Y \to T$, $T \to X$, $X—Z$, $Z—Y \in E_e$, $Y \to Z \in E_a$ y $X \to Z \in E_o$, las restricciones de ausencia y de orden obligan la orientación de las aristas $Z—Y$ y $X—Z$, las cuales, junto con la otra restricción de existencia, generan un ciclo dirigido. El siguiente resultado caracteriza la autoconsistencia *global* de las restricciones, en términos de simples operaciones sobre grafos.

**Proposición 3.5.2** *Sean $G_e = (\mathbf{X}, E_e)$, $G_a = (\mathbf{X}, E_a)$ y $G_o = (\mathbf{X}, E_o)$ los grafos que representan las restricciones de existencia, ausencia y orden, respectivamente. Sea $G_{re} = (\mathbf{X}, E_{re})$ el grafo refinado de restricciones de exis-*

*tencia*[2] *definido como*

$$E_{re} = \{X \to Y \mid X \to Y \in E_e\} \cup \{Y \to X \mid X — Y \in E_e,\ X \to Y \in E_a\} \cup \{X — Y \mid X — Y \in E_e,\ X \to Y \notin E_a,\ Y \to X \notin E_a\} \tag{3.1}$$

*Entonces los tres conjuntos de restricciones son autoconsistentes si y sólo si* $G_{re} \cap G_a = G_\emptyset$ *y* $G_{re} \cup G_o$ *no tiene ningún ciclo dirigido.* $G_\emptyset$ *es un grafo vacío*[3] *y tanto la unión como la intersección de dos grafos parcialmente dirigidos usan la siguiente convención* $\{X \to Y\} \cup \{X — Y\} = \{X \to Y\}$ *y* $\{X \to Y\} \cap \{X — Y\} = \{X \to Y\}$.

**Demostración.** *Condición necesaria*: sabemos que existe un DAG $G$ consistente con las restricciones. Primero, veamos que $G_{re} \cap G_a = G_\emptyset$:

$X \to Y \in E_{re}$ si y sólo si $X \to Y \in E_e$ o $X — Y \in E_e$ y $Y \to X \in E_a$. En el primer caso, de $X \to Y \in E_e$ tomamos $X \to Y \in E_G$ y de este $X \to Y \notin E_a$ y $X — Y \notin E_a$. En el segundo caso, de $X — Y \in E_e$ tomamos $X \to Y \in E_G$ o $Y \to X \in E_G$; si $X \to Y \in E_G$ entonces también obtenemos $X \to Y \notin E_a$ y $X — Y \notin E_a$; el segundo caso, $Y \to X \in E_G$, no puede darse porque también tenemos que $Y \to X \in E_a$. Por lo tanto, tenemos que si $X \to Y \in E_{re}$ entonces $X \to Y \notin E_a$ y $X — Y \notin E_a$. Por otro lado, $X — Y \in E_{re}$ si y sólo si $X — Y \in E_e$, $X \to Y \notin E_a$ y $Y \to X \notin E_a$. De $X — Y \in E_e$ obtenemos que $X \to Y \in E_G$ o $Y \to X \in E_G$, y en cualquier caso esto implica que $X — Y \notin E_a$. De esta forma, tenemos que si $X — Y \in E_{re}$ entonces $X \to Y \notin E_a$, $Y \to X \notin E_a$ y $X — Y \notin E_a$. Por lo tanto, $G_{re} \cap G_a = G_\emptyset$.

Ahora, veamos que $G_{re} \cup G_o$ no tiene ciclos dirigidos:

Si $G_{re} \cup G_o$ tiene algún ciclo dirigido $X_1 \to X_2 \to \ldots \to X_k \to X_1$, entonces cada uno de estos arcos pertenece a $E_{re}$ o $E_o$. Si $X_i \to X_{i+1} \in E_o$ entonces todo orden total $\sigma$ compatible con $G_o$ tiene que verificar que $X_i <_\sigma X_{i+1}$. Si $X_i \to X_{i+1} \in E_{re}$ entonces $X_i \to X_{i+1} \in E_G$ y de nuevo todo orden total $\sigma$ compatible con $G$ tiene que verificar que $X_i <_\sigma X_{i+1}$. En consecuencia, todo orden total compatible con los dos grafos $G$ y $G_o$ debe verificar que $X_1 <_\sigma X_2 <_\sigma \ldots <_\sigma X_k <_\sigma X_1$, lo cual es obviamente imposible. Por lo tanto $G_{re} \cup G_o$ no tiene ciclos dirigidos.

*Condición suficiente*: sabemos que $G_{re} \cap G_a = G_\emptyset$ y $G_{re} \cup G_o$ no tiene ciclos dirigidos. Como $G_{re} \cup G_o$ es un grafo sin ciclos dirigidos podemos orientar las

[2] $G_{re}$ es el grafo $G_e$ con algunas aristas sustituidas por arcos (justo aquellas cuya dirección está forzada por alguna restricción de ausencia).

[3] un grafo que no tiene ni arcos ni aristas.

aristas de este grafo para obtener un DAG. Seleccionamos solamente los arcos de este DAG completo que procede de $G_{re}$, y obtenemos un nuevo subDAG $G = (\mathbf{X}, E_G)$. Obviamente $G$ satisface las restricciones de existencia. Veamos que $G$ también verifica las restricciones de ausencia y orden: Si $X \rightarrow Y \in E_a$ entonces $X \rightarrow Y \notin E_{re}$ y $X—Y \notin E_{re}$, por lo tanto $X \rightarrow Y \notin E_G$. Si $X—Y \in E_a$ entonces $X—Y \notin E_{re}$, $X \rightarrow Y \notin E_{re}$ y $Y \rightarrow X \notin E_{re}$, por lo tanto $X \rightarrow Y \notin E_G$ y $Y \rightarrow X \notin E_G$, y $G$ satisface las restricciones de ausencia. Como $G \cup G_o$ es un DAG entonces existe un orden compatible con $G \cup G_o$. Este orden es claramente compatible con $G$ y $G_o$, y entonces $G$ satisface las restricciones de orden. ∎

## 3.6. Verificación de restricciones usando operaciones sobre grafos.

El siguiente resultado muestra que comprobar la consistencia de un DAG con un conjunto de restricciones también puede reducirse a simples operaciones sobre grafos.

**Proposición 3.6.1** *Sean $G_e = (\mathbf{X}, E_e)$, $G_a = (\mathbf{X}, E_a)$ y $G_o = (\mathbf{X}, E_o)$ los grafos que representan los conjuntos autoconsistentes de restricciones de existencia, ausencia y orden, respectivamente, y sea $G = (\mathbf{X}, E_G)$ un DAG. Entonces $G$ es consistente con las restricciones si y sólo si $G \cup G_e = G$, $G \cap G_a = G_\emptyset$ y $G \cup G_o$ es un DAG.*

**Demostración.** *Condición necesaria*: sabemos que $G$ es consistente con las restricciones. Primero demostremos que $G \cup G_e = G$. Si $X \rightarrow Y \in E_G \cup E_e$ entonces $X \rightarrow Y \in E_G$ o $X \rightarrow Y \in E_e$. En el segundo caso podemos deducir que $X \rightarrow Y \in E_G$. Si $X—Y \in E_G \cup E_e$ entonces $X—Y \in E_e$, $X \rightarrow Y \notin E_G$ y $Y \rightarrow X \notin E_G$, pero esta situación no es posible porque $G$ es consistente con $G_e$. Por lo tanto obtenemos que $G \cup G_e \subseteq G$, y obviamente $G \subseteq G \cup G_e$.

Ahora, demostremos que $G \cap G_a = G_\emptyset$. Si $X \rightarrow Y \in E_a$ entonces $X \rightarrow Y \notin E_G$. Si $X—Y \in E_a$ entonces $X \rightarrow Y \notin E_G$ y $Y \rightarrow X \notin E_G$. Por lo tanto $G \cap G_a = G_\emptyset$.

Finalmente, probemos que $G \cup G_o$ es un DAG. En el caso de que $G \cup G_o$ no sea un DAG, entonces existe un ciclo dirigido $X_1 \rightarrow X_2 \rightarrow \ldots \rightarrow X_k \rightarrow X_1$, cada uno de estos arcos pertenecen a $E_G \cup E_o$. Entonces, igual que en la demostración de la proposición 3.5.2 para demostrar que $G_{re} \cup G_o$ no tenía

ciclos dirigidos, podemos deducir que todo orden total $\sigma$ compatible con $G$ y $G_o$ debe verificar que $X_1 <_\sigma X_2 <_\sigma \ldots <_\sigma X_k <_\sigma X_1$.

*Condición suficiente*: sabemos que $G \cup G_e = G$, $G \cap G_a = G_\emptyset$ y $G \cup G_o$ es un DAG. Como $E_G \cup E_e = E_G$, si $X \rightarrow Y \in E_e$ entonces $X \rightarrow Y \in E_G$; si $X—Y \in E_e$ entonces $X \rightarrow Y \in E_G$ o $Y \rightarrow X \in E_G$. Por lo tanto $G$ es consistente con las restricciones de existencia.

Como $E_a \cap E_G = \emptyset$, si $X \rightarrow Y \in E_a$ entonces $X \rightarrow Y \notin E_G$; si $X—Y \in E_a$ entonces $X \rightarrow Y \notin E_G$ y $Y \rightarrow X \notin E_G$, y $G$ es consistente con las restricciones de ausencia.

Como $G \cup G_o$ es un DAG, existe un orden total $\sigma$ compatible con $G \cup G_o$. Entonces este orden es compatible con $G$ y $G_o$. ∎

Comprobar la consistencia de un PDAG con un conjunto de restricciones es sólo un poco más complicado, debido a la posible interacción entre las aristas del PDAG y los arcos de las restricciones de ausencia.

**Proposición 3.6.2** *Sean $G_e = (\mathbf{X}, E_e)$, $G_a = (\mathbf{X}, E_a)$ y $G_o = (\mathbf{X}, E_o)$ los grafos que representan los conjuntos autoconsistentes de restricciones de existencia, ausencia y orden, respectivamente, y sea $H = (\mathbf{X}, E_H)$ un PDAG y sea $H_r = (\mathbf{X}, E_{Hr})$ el PDAG refinado[4] definido como*

$$\begin{aligned} E_{Hr} \;=\; & \{X \rightarrow Y \mid X \rightarrow Y \in E_H\} \cup \{Y \rightarrow X \mid X—Y \in E_H,\, X \rightarrow Y \in E_a\} \cup \\ & \{X—Y \mid X—Y \in E_H,\, X \rightarrow Y \notin E_a,\, Y \rightarrow X \notin E_a\} \end{aligned} \tag{3.2}$$

*Entonces $H$ es consistente con las restricciones si y sólo si $H \cup G_e = H$, $H_r \cap G_a = G_\emptyset$ y $H_r \cup G_o$ no tiene ningún ciclo dirigido.*

**Demostración.** Esta demostración es bastante similar a la demostración de las proposiciones 3.5.2 y 3.6.1, así que vamos a omitir los detalles. La justificación para usar $H_r$ en lugar de $H$ es que, como sucedía con las restricciones de existencia $G_e$ en la proposición 3.5.2, las aristas en $H$ pueden interaccionar con los arcos $G_a$ (por ejemplo, si $E_a = \{X_1 \rightarrow X_2\}$ y $E_H = \{X_1—X_2\}$, entonces $G_a \cap H \neq G_\emptyset$ porque $\{X_1 \rightarrow X_2\} \cap \{X_1—X_2\} = \{X_1 \rightarrow X_2\}$). Las aristas de $H$ cuya orientación es forzada por los arcos de $G_a$ pueden también interactuar con los arcos de $G_o$ (por ejemplo, si $E_a = \{X_1 \rightarrow X_2, X_2 \rightarrow X_3\}$ y $E_o =$

[4]Como antes, $H_r$ es el grafo $H$ donde a algunas aristas se les ha dado una dirección, debido a una restricción de ausencia.

$\{X_1 \rightarrow X_3\}$, entonces el grafo $H = (\mathbf{X}, E_H)$ con $E_H = \{X_1—X_2, X_2—X_3\}$, verifica que $H \cup G_o$ no tiene ciclos dirigidos aunque $H$ no es consistente con las restricciones, mientras que $H_r \cup G_o$ sí tiene un ciclo dirigido). ∎

## 3.7. Restricciones en métodos métrica+búsqueda.

Si tenemos un conjunto de restricciones autoconsistentes y queremos construir una red bayesiana a partir de un conjunto de datos usando, para ello, un algoritmo de métrica+búsqueda, parece natural usar las restricciones para reducir el espacio de búsqueda y forzar al algoritmo a devolver un DAG consistente con las restricciones. Un mecanismo general para hacerlo, el cual es válido para cualquier algoritmo de este tipo de metodologías, es muy simple: cada vez que el proceso de búsqueda selecciona un DAG candidato $G$ para que sea evaluado por la métrica, podemos usar el resultado de la proposición 3.6.1 para comprobar si $G$ es consistente con las restricciones, y descartarlo en caso negativo.

No obstante, este procedimiento general puede ser algo ineficiente. Sería conveniente adaptarlo a las características específicas del algoritmo de aprendizaje usado. Lo vamos a hacer para el caso del algoritmo de métrica+búsqueda basado en búsqueda local y presentado en [150], el cual utiliza operadores de inserción de arcos, eliminación de arcos e inversión de arcos.

### 3.7.1. Búsqueda local.

Antes de hacer uso de restricciones, veamos cómo sería el algoritmo de búsqueda local para el aprendizaje de la estructura de una red bayesiana. Para ello, tengamos en cuenta que muchos procedimientos de búsqueda, incluyendo la búsqueda local, se basan en el concepto de vecindad, donde nos podemos mover de una estructura vecina a otra mediante la aplicación de tres operadores sobre arcos: inserción, eliminación o inversión[5], teniendo en cuenta que tanto la inserción de un arco como la inversión deben comprobar que no se formen ciclos.

[5]Incluso podrían considerarse sólo dos operadores: inserción y eliminación, puesto que el operador de inversión se puede ver como una eliminación más una inserción.

Los algoritmos de búsqueda en el espacio de DAGs, que utilizan los mencionados operadores, resultan ser bastante eficientes debido principalmente a la descomponibilidad de muchas métricas. Una métrica $f$ se dice que es *descomponible* si la medida de cualquier estructura de red bayesiana puede ser expresada en base a las medidas locales involucrando sólo un nodo y a sus padres:

$$f(G : \mathcal{D}) = \sum_{X \in \mathbf{X}} f_{\mathcal{D}}(X, Pa_G(X))$$

$$f_{\mathcal{D}}(X, Pa_G(X)) = f(X, Pa_G(X) : N_{X,Pa_G(X)})$$

donde $Pa_G(X)$ es el conjunto de padres del nodo $X$, es decir, $Pa_G(X) = \{Y \in \mathbf{X} | Y \rightarrow X \in G\}$, y $N_{X,Pa_G(X)}$ es el número de instancias de $\mathcal{D}$ en que aparecen conjuntamente los distintos valores de $X$ y $Pa_G(X)$.

Un algoritmo que en cada movimiento va cambiando un sólo arco (inserción, eliminación o inversión) puede evaluar de manera eficiente la mejora obtenida mediante este cambio. De esta forma se puede reutilizar el valor de la métrica calculada en pasos anteriores y calcular solamente el valor para aquella variable y sus padres donde se produzca un cambio.

Por tanto, la inserción o eliminación de un arco $X \rightarrow Y$ en un DAG $G$ se puede evaluar con sólo una medida local, $f_{\mathcal{D}}(Y, Pa_G(Y) \cup \{X\})$ o $f_{\mathcal{D}}(Y, Pa_G(Y) \setminus \{X\})$, respectivamente. La evaluación de la inversión de un arco $X \rightarrow Y$ necesita el cálculo de dos medidas locales $f_{\mathcal{D}}(Y, Pa_G(Y) \setminus \{X\})$ y $f_{\mathcal{D}}(X, Pa_G(X) \cup \{Y\})$.

Teniendo en cuenta que a cualquier estructura vecina se llega mediante la aplicación de los operadores mencionados, el proceso de búsqueda local se puede ver en el Algoritmo 9.

**Algoritmo 9** Búsqueda local en la estructura de una red bayesiana.

1: Comenzamos con un DAG $G$ (normalmente $G = G_\emptyset$)
2: Calculamos el valor de la métrica para $G$, como $f(G : \mathcal{D})$
3: **repite**
4: **para cada** par $\{X, Y \in \mathbf{X} | X \rightarrow Y \notin G, Y \rightarrow X \notin G\}$ **repite**
5: Calcula $Diferencia = f_\mathcal{D}(Y, Pa_G(Y) \cup \{X\}) - f_\mathcal{D}(Y, Pa_G(Y))$ {Insertar arco}
6: **fin para**
7: **para cada** enlace $X \rightarrow Y \in G$ **repite**
8: Calcula $Diferencia = f_\mathcal{D}(Y, Pa_G(Y) \setminus \{X\}) - f_\mathcal{D}(Y, Pa_G(Y))$ {Eliminar arco}
9: Calcula $Diferencia = f_\mathcal{D}(Y, Pa_G(Y) \setminus \{X\}) - f_\mathcal{D}(Y, Pa_G(Y)) + f_\mathcal{D}(X, Pa_G(X) \cup \{Y\}) - f_\mathcal{D}(X, Pa_G(X))$ {Invertir arco}
10: **fin para**
11: Obtener la $Diferencia$ máxima de entre todas las calculadas y guardar en $Diferencia_{max}$
12: **si** $(Diferencia_{max} > 0)$ **entonces**
13: Aplicar sobre $G$ la operación de inserción, eliminación o inversión relacionada con $Diferencia_{max}$
14: $f(G : \mathcal{D}) = f(G : \mathcal{D}) + Diferencia_{max}$
15: **fin si**
16: **hasta que** $(Diferencia_{max} \leq 0)$

### 3.7.2. Condiciones que se aplican a los operadores de búsqueda.

Comenzamos con el actual DAG $G$, el cual es consistente con las restricciones, y sea $G'$ el DAG obtenido a partir de $G$ mediante la aplicación de los operadores de inserción de arcos, eliminación de arcos e inversión de arcos. Veamos qué condiciones son necesarias y suficientes para asegurar que $G'$ es también consistente con las restricciones

**Proposición 3.7.1** *Sean $G_e = (\mathbf{X}, E_e)$, $G_a = (\mathbf{X}, E_a)$ y $G_o = (\mathbf{X}, E_o)$ los grafos que representan las restricciones autoconsistentes de existencia, ausencia y orden, respectivamente, y sea $G = (\mathbf{X}, E_G)$ un DAG consistente con las restricciones anteriores.*

(a) *Inserción: sea* $G' = (\mathbf{X}, E'_G)$, $E'_G = E_G \cup \{X \rightarrow Y\}$, *con* $X \rightarrow Y \notin E_G$. *Entonces* $G'$ *es un DAG consistente con las restricciones si y sólo si*

- $X \rightarrow Y \notin E_a$ *y* $X$—$Y \notin E_a$,
- *no hay ningún camino dirigido desde* $Y$ *a* $X$ *en* $G \cup G_o$.

(b) *Eliminación: sea* $G' = (\mathbf{X}, E'_G)$, $E'_G = E_G \setminus \{X \rightarrow Y\}$, *con* $X \rightarrow Y \in E_G$. *Entonces* $G'$ *es un DAG consistente con las restricciones si y sólo si*

- $X \rightarrow Y \notin E_e$ *y* $X$—$Y \notin E_e$.

(c) *Inversión: sea* $G' = (\mathbf{X}, E'_G)$, $E'_G = (E_G \setminus \{X \rightarrow Y\}) \cup \{Y \rightarrow X\}$, *con* $X \rightarrow Y \in E_G$. *Entonces* $G'$ *es un DAG consistente con las restricciones si y sólo si*

- $X \rightarrow Y \notin E_e$, $Y \rightarrow X \notin E_a$ *y* $X \rightarrow Y \notin E_o$,
- *excluyendo al arco* $X \rightarrow Y$, *no hay ningún otro camino dirigido de* $X$ *a* $Y$ *en* $G \cup G_o$.

**Demostración.** (a) *Condición necesaria*: sabemos que $G'$ es consistente con las restricciones. Como $E'_G \cap E_a = \emptyset$, entonces $(E_G \cup \{X \rightarrow Y\}) \cap E_a = \emptyset$. Por lo tanto $\{X \rightarrow Y\} \cap E_a = \emptyset$ y esto significa que $X \rightarrow Y \notin E_a$ y $X$—$Y \notin E_a$. Por otro lado, si $G \cup G_o$ tuviera un camino dirigido de $Y$ a $X$, entonces deberíamos tener un ciclo en $G' \cup G_o$, y entonces $G' \cup G_o$ no sería un DAG.

*Condición suficiente*: sabemos que $G$ es consistente con las restricciones,

Como $E_G \cup E_e = E_G$, entonces $E'_G \cup E_e = (E_G \cup \{X \rightarrow Y\}) \cup E_e = E_G \cup \{X \rightarrow Y\} = E'_G$ y por lo tanto $G' \cup G_e = G'$.

Como $E_G \cap E_a = \emptyset$, entonces $E'_G \cap E_a = (E_G \cup \{X \rightarrow Y\}) \cap E_a = \{X \rightarrow Y\} \cap E_a$, y esta última intersección está vacía debido a que $X \rightarrow Y \notin E_a$ y $X$—$Y \notin E_a$. Por lo tanto $G' \cap G_a = G_\emptyset$.

Finalmente, como $G \cup G_o$ es un DAG y no hay un camino dirigido de $Y$ a $X$ en este grafo, entonces el grafo $G' \cup G_o$ obtenido de $G \cup G_o$ al incluir el arco $X \rightarrow Y$ es un DAG.

(b) *Condición necesaria*: sabemos que $G$ y $G'$ son consistentes con las restricciones. Como $G' \cup G_e = G'$, entonces $E'_G \cup E_e = E'_G$, es decir, $(E_G \setminus \{X \rightarrow Y\}) \cup E_e = E_G \setminus \{X \rightarrow Y\}$. En caso de que $X \rightarrow Y \in E_e$ entonces tendríamos $(E_G \setminus \{X \rightarrow Y\}) \cup E_e = E_G \cup E_e = E_G$. En caso de que $X$—$Y \in E_e$ entonces

$(E_G \setminus \{X \to Y\}) \cup E_e = ((E_G \cup E_e) \setminus \{X \to Y\}) \cup \{X—Y\} = (E_G \setminus \{X \to Y\}) \cup \{X—Y\}$, y ambos casos contradicen la hipótesis.

*Condición suficiente*: como $E_G \cap E_a = \emptyset$ entonces $E'_G \cap E_a = (E_G \setminus \{X \to Y\}) \cap E_a = \emptyset$, y $G' \cap G_a = G_\emptyset$.

Como $E_G \cup E_e = E_G$, $X \to Y \notin E_e$ y $X—Y \notin E_e$, entonces $E'_G \cup E_e = (E_G \setminus \{X \to Y\}) \cup E_e = (E_G \cup E_e) \setminus \{X \to Y\} = E_G \setminus \{X \to Y\} = E'_G$. Así, $G' \cup G_e = G'$.

Finalmente, como $G \cup G_o$ es un DAG y $G' \cup G_o$ es un subgrafo, es también un DAG.

(c) *Condiciones necesarias y suficientes* : invirtiendo un arco $X \to Y$ puede verse como un borrado de dicho arco y una inserción posterior del arco $Y \to X$. Entonces, aplicando las condiciones de eliminación e inserción tendríamos $X \to Y \notin E_e$, $X—Y \notin E_e$, $Y \to X \notin E_a$, $X—Y \notin E_a$ y no hay ningún camino dirigido de $X$ a $Y$ en $(G \setminus \{X \to Y\}) \cup G_o$. No obstante, la condición $X—Y \notin E_e$ no es necesaria, debido a que no estamos eliminado el arco $X \to Y$ lo estamos sustituyendo por $Y \to X$. La condición $X—Y \notin E_a$ siempre será verdad, porque $X \to Y$ estaba en $G$ y $G$ es consistente. Finalmente, la condición que indicaba que no hay ningún camino dirigido de $X$ a $Y$ en $(G \setminus \{X \to Y\}) \cup G_o$ es equivalente a decir que no hay ningún camino dirigido de $X$ a $Y$ en $(G \cup G_o) \setminus \{X \to Y\}$ y que $X \to Y \notin E_o$. ∎

Tengamos en cuenta que las condiciones sobre la ausencia de caminos dirigidos entre $X$ e $Y$ en la proposición previa también tienen que ser comprobadas por un algoritmo que no considere las restricciones (usando en este caso el DAG $G$ en lugar de $G \cup G_o$), por lo tanto el costo computacional extra por el uso de las restricciones es bastante reducido: sólo dos o tres comprobaciones sobre la ausencia de un arco o una arista en el grafo.

También es interesante darnos cuenta de que otros algoritmos de aprendizaje basados en el paradigma métrica+búsqueda, que sean más sofisticados que una simple búsqueda local, también pueden ser extendidos de forma eficiente para que trabajen con restricciones. Hay muchos algoritmos de aprendizaje de redes bayesianas que realizan una búsqueda más exhaustiva que la búsqueda local pero usan los mismos tres operadores básicos (como en búsquedas de entorno variable [90], búsqueda tabú [6] o en procedimientos de búsquedas voraces y aleatorios [87]) o, incluso, un subconjunto de estos operadores (como en colonias de hormigas [86], que sólo usa el operador de

inserción[6]). Todos estos algoritmos pueden ser usados con restricciones sin apenas modificaciones.

### 3.7.3. Inicialización de la búsqueda.

Otra cuestión a considerar es la inicialización del proceso de búsqueda. En general, los algoritmos de búsqueda empiezan a partir de uno o varios DAGs iniciales que, en nuestro caso, deben ser consistentes con las restricciones. Un punto de partida muy común es un DAG vacío $G_\emptyset$. En nuestro caso $G_\emptyset$ debe ser sustituido por el grafo $G_e$ o, incluso mejor, por el grafo refinado de existencia, $G_{re}$. No obstante, como $G_{re}$ no es necesariamente un DAG, debe ser convertido en un DAG. Una forma fácil de hacerlo es seleccionar iterativamente una arista $X—Y \in E_{re}$, aleatoriamente escoger una orientación y comprobar si las restricciones son autoconsistentes, escogiendo la orientación contraria si no lo son. Este proceso está basado en el siguiente resultado:

**Proposición 3.7.2** *Sean $G_e = (\mathbf{X}, E_e)$, $G_a = (\mathbf{X}, E_a)$ y $G_o = (\mathbf{X}, E_o)$ los grafos autoconsistentes que representan las restricciones de existencia, ausencia y orden, respectivamente. Sea $G_{re} = (\mathbf{X}, E_{re})$ el grafo refinado de las restricciones de existencia. Sea $X—Y \in E_{re}$ y definamos el grafo $G_{e(X\to Y)} = (\mathbf{X}, E_{e(X\to Y)})$, donde $E_{e(X\to Y)} = (E_e \setminus \{X—Y\}) \cup \{X \to Y\}$. Entonces $G_{e(X\to Y)}$, $G_a$ y $G_o$ son aún autoconsistentes si y sólo si no hay un camino dirigido desde $Y$ a $X$ en $G_{re} \cup G_o$. Por otro lado, bien $G_{e(X\to Y)}$ o $G_{e(Y\to X)}$, junto con $G_a$ y $G_o$, son autoconsistentes.*

**Demostración.** Sea $G_{re(X\to Y)} = (\mathbf{X}, E_{re(X\to Y)})$ el grafo refinado de las restricciones de existencia, usando $G_{e(X\to Y)}$ en lugar de $G_e$. Está claro que $E_{re(X\to Y)} = (E_{re} \setminus \{X—Y\}) \cup \{X \to Y\}$. Como $X—Y \in E_{re}$ también sabemos que $X \to Y \notin E_a$ y $Y \to X \notin E_a$. Entonces, de acuerdo a la proposición 3.5.2, $G_{e(X\to Y)}$, $G_a$ y $G_o$ serán autoconsistentes si y sólo si $G_{re(X\to Y)} \cap G_a = G_\emptyset$ y $G_{re(X\to Y)} \cup G_o$ no tienen un ciclo dirigido.

Asumamos primero que $G_{e(X\to Y)}$, $G_a$ y $G_o$ son autoconsistentes. Si hay un camino dirigido de $Y$ a $X$ en $G_{re} \cup G_o$, todos los arcos del camino están también en $G_{re(X\to Y)} \cup G_o$ y, junto con el arco $X \to Y$, forman un ciclo dirigido en $G_{re(X\to Y)} \cup G_o$, lo cual contradice la hipótesis.

[6]El algoritmo B [56] también sólo usa el operador de inserción, junto con búsqueda local.

Ahora, vamos a asumir que no hay ningún camino dirigido de $Y$ a $X$ en $G_{re} \cup G_o$. Como $G_e$, $G_a$ y $G_o$ son autoconsistentes, también sabemos que $G_{re} \cap G_a = G_\emptyset$ y $G_{re} \cup G_o$ no tienen un ciclo dirigido.

$E_{re(X \to Y)} \cap E_a = ((E_{re} \setminus \{X — Y\}) \cup \{X \to Y\}) \cap E_a = ((E_{re} \setminus \{X — Y\}) \cap E_a) \cup (\{X \to Y\} \cap E_a) = \emptyset$. Por lo tanto, $G_{re(X \to Y)} \cap G_a = G_\emptyset$.

Como $G_{re} \cup G_o$ no tiene un ciclo dirigido, si después de orientar la arista $X — Y$ como $X \to Y$, para obtener $G_{re(X \to Y)} \cup G_o$, obtenemos un ciclo, entonces hay un camino dirigido de $Y$ a $X$ en $G_{re} \cup G_o$, lo cual contradice la hipótesis. Por lo tanto, tenemos que $G_{re(X \to Y)} \cap G_a = G_\emptyset$ y $G_{re(X \to Y)} \cup G_o$ no tiene un ciclo dirigido, por lo que $G_{e(X \to Y)}$, $G_a$ y $G_o$ son autoconsistentes.

Finalmente, demostremos que $G_{e(X \to Y)}$ o $G_{e(Y \to X)}$, junto con $G_a$ y $G_o$ son autoconsistentes. Si asumimos que esto no es verdad, entonces hay un camino dirigido de $Y$ a $X$ y otro de $X$ a $Y$ en $G_{re} \cup G_o$, y esto significa que tenemos un ciclo dirigido en $G_{re} \cup G_o$, lo cual contradice el hecho de que $G_e$, $G_a$ y $G_o$ son autoconsistentes. ∎

En otros casos el algoritmo de búsqueda es inicializado con uno (o varios) DAGs aleatorios. El proceso de seleccionar un DAG aleatorio, comprobando las restricciones e iterando hasta que el DAG generado satisfaga las restricciones, puede ser demasiado costoso en tiempo, especialmente cuando tengamos muchas restricciones. En estos casos podría ser útil un operador reparador, es decir, un método que transforme cualquier DAG $G$ en uno que verifique las restricciones. Este método puede ser también útil para algoritmos de aprendizaje que usan un algoritmo de búsqueda basado en poblaciones (como los algoritmos genéticos [209] y los EDAs [43]). Hay muchas formas de definir este operador de reparación. Aquí proponemos un método bastante simple: empezamos con un DAG $G_{red}$ conteniendo sólo los arcos[7] (no las aristas) en $G_{re}$; entonces, dado un orden aleatorio de los arcos de $G$, iterativamente intentamos insertar cada uno de esos arcos en $G_{red}$, usando las condiciones de la proposición 3.7.1(a); finalmente, para las aristas de $G_{re}$, las incluimos en $G_{red}$ con la orientación apropiada, usando la comprobación de la proposición 3.7.2 (sustituyendo el grafo $G_{re} \cup G_o$ por $G_{red} \cup G_o$). El resultado es un DAG consistente con las restricciones que contiene tantos arcos de $G$ como es posible.

[7] $G_{red} = (\mathbf{X}, E_{red})$, con $E_{red} = \{X \to Y \mid X \to Y \in E_{re}\}$.

## 3.8. Restricciones en métodos basados en tests de independencia.

La forma de proceder normalmente en los algoritmos de aprendizaje basados en tests de independencia es, primero, eliminando aristas que conectan pares de nodos que son condicionalmente independientes dado algún subconjunto de nodos y, segundo, orientando aristas a patrones cabeza-cabeza (tripletas de nodos $X, Y, Z$ donde $X$ e $Y$ no son adyacentes y los arcos $X \rightarrow Z$ y $Y \rightarrow Z$ existen). Ambas actividades son guiadas por los resultados de tests de independencia condicional aplicados a los datos disponibles. Por ejemplo, los algoritmos SGS y PC [321] eliminan primero tantas aristas como puedan, y después direccionan algunas de las aristas que no se pueden eliminar, formando patrones cabeza-cabeza. Finalmente se les da una dirección al resto de las aristas que queden usando algunas reglas de coherencia.

Al aplicar restricciones sobre este tipo de algoritmos de aprendizaje estructural de redes bayesianas, el objetivo perseguido es utilizar las restricciones para disminuir el número de tests de independencia a realizar. Téngase en cuenta que detectar independencias condicionales de orden elevado (donde intervienen muchas variables) es bastante costoso computacionalmente.

### 3.8.1. El algoritmo PC.

El uso de las restricciones en el caso de los métodos basados en tests de independencia lo vamos aplicar al caso concreto del algoritmo PC, por ser uno de los más conocidos y utilizados dentro de este tipo de métodos. El algoritmo PC presupone que el modelo que se pretende recuperar es isomorfo a un grafo dirigido acíclico, es decir, todas las relaciones de independencia condicional del modelo se corresponden con relaciones de independencia gráfica o d-separación de la estructura, y viceversa.

Comienza con un grafo completo no dirigido para, posteriormente, ir reduciéndolo. Primero elimina las aristas que unen dos nodos que verifican una independencia condicional de orden cero, después las de orden uno, y así sucesivamente. El conjunto de nodos candidatos para formar el conjunto separador (el conjunto al que se condiciona) es el de los nodos adyacentes

a alguno de los nodos que se pretenden separar. Los pasos se detallan en el algoritmo 10.

**Algoritmo 10** Algoritmo PC.

1: Formar el grafo completo no dirigido $G$
2: $n = 0$
3: **repite**
4: **repite**
5: Seleccionar un par de nodos $X, Y$ adyacentes en $G$ tales que $|Ady_G(X, Y)| \geq n$ ($Ady_G(X, Y)$ son los nodos adyacentes a $X$ o a $Y$), y seleccionar un subconjunto $S(X, Y) \subseteq Ady_G(X, Y)$ de cardinal igual a $n$.
6: Si $I(X, Y|S(X, Y))$, eliminar $X$—$Y$ de $G$, y guardar $S(X, Y)$
7: **hasta que** todos los pares $X, Y$ y todos los subconjuntos $S(X, Y)$ hayan sido comprobados
8: $n = n + 1$
9: **hasta que** para cada par de nodos adyacentes $X, Y, |Ady_G(X, Y)| < n$
10: Sea $G$ el grafo resultante de los pasos anteriores. Para cada terna $X, Y, Z$ tal que $X$—$Y$—$Z \in G$, pero $X$—$Z \notin G$, orientar como $X \rightarrow Z \leftarrow Y$ si y sólo si $Y \notin S(X, Z)$

Como todos los algoritmos que recuperan grafos generales, en el peor de los casos posibles, la complejidad del algoritmo PC es exponencial, aunque es razonablemente eficiente para redes poco densas.

### 3.8.2. Condiciones en las que se aplican los tests de independencia.

Sin entrar en mucho detalle, una forma simple de usar un conjunto de restricciones con el objetivo de reducir el número de tests necesarios, es la siguiente: antes de aplicar un test de independencia $I(X, Y \mid Z)$, para eliminar una arista $X$—$Y$ o crear un patrón cabeza-cabeza $X \rightarrow Z \leftarrow Y$, debemos comprobar si el grafo obtenido tras la aplicación de esta operación es consistente con las restricciones (usando el resultado de la proposición 3.6.2); si el test de consistencia falla, el test de independencia no se llevará a cabo.

No obstante, para mejorar la eficiencia del algoritmo, es aconsejable adaptar el test de consistencia general presentado en la proposición 3.6.2 a las características específicas de los operadores que se estén usando en los métodos basados en tests de independencia.

**Proposición 3.8.1** *Sean $G_e = (\mathbf{X}, E_e)$, $G_a = (\mathbf{X}, E_a)$ y $G_o = (\mathbf{X}, E_o)$ los grafos que representan restricciones autoconsistentes de existencia, ausencia y orden, respectivamente, y sea $H = (\mathbf{X}, E_H)$ un PDAG consistente con las restricciones.*

*(a) Eliminación de arcos: sea $H' = (\mathbf{X}, E'_H)$, $E'_H = E_H \setminus \{X \to Y\}$, con $X \to Y \in E_H$. Entonces $H'$ es un PDAG consistente con las restricciones si y sólo si*

- *$X \to Y \notin E_e$ y $X — Y \notin E_e$.*

*(b) Eliminación de aristas: sea $H' = (\mathbf{X}, E'_H)$, $E'_H = E_H \setminus \{X — Y\}$, con $X — Y \in E_H$. Entonces $H'$ es un PDAG consistente con las restricciones si y sólo si*

- *$X \to Y \notin E_e$, $Y \to X \notin E_e$ y $X — Y \notin E_e$.*

*(c) Inserción patrón cabeza-cabeza: sea $X, Y, Z \in \mathbf{X}$ y definamos un subconjunto de enlaces $S$ como $S = \{X \to Z, Z — Y\}$, $S = \{X — Z, Y \to Z\}$ o $S = \{X — Z, Z — Y\}$. Si $X$ e $Y$ no son adyacentes en $H$ y $S \subseteq E_H$, sea $H' = (\mathbf{X}, E'_H)$, con $E'_H = (E_H \setminus S) \cup \{X \to Z, Y \to Z\}$. Entonces $H'$ es un PDAG consistente con las restricciones si y sólo si*

- *$X \to Z \notin E_a$ y $Y \to Z \notin E_a$,*
- *no hay un camino dirigido de $Z$ a $X$ ni de $Z$ a $Y$ en $H \cup G_o$.*

*(d) Orientación de aristas: sea $H' = (\mathbf{X}, E'_H)$, $E'_H = (E_H \setminus \{X — Y\}) \cup \{X \to Y\}$, con $X — Y \in E_H$. Entonces $H'$ es un PDAG consistente con las restricciones si y sólo si*

- *$X \to Y \notin E_a$,*
- *no hay un camino dirigido de $Y$ a $X$ en $H \cup G_o$.*

**Demostración.** (a) y (b) La demostración es completamente similar a la de la proposición 3.7.1(b). La diferencia radica en que usamos $H_r$ en lugar de $H$, pero solamente tenemos que tener en cuenta que $H'_r$ es siempre un subgrafo de $H_r$.

(d) Orientar una arista $X$—$Y$ es similar a insertar un arco $X \rightarrow Y$, así que las condiciones que aseguran la consistencia son las de la proposición 3.7.1(a); la única diferencia es que la condición $X$—$Y \notin E_a$ en la proposición 3.7.1(a) no es necesaria (es siempre verdad), ya que la arista $X$—$Y$ ya está en $H$ y $H$ es consistente con las restricciones.

(c) Como en el caso previo, crear patrones cabeza-cabeza es similar a insertar dos arcos, así que las condiciones para la consistencia están otra vez en la proposición 3.7.1(a) aplicada a los arcos $X \rightarrow Z$ y $Y \rightarrow Z$; como antes, sabemos que hay enlaces uniendo los nodos $X$ y $Z$ y los nodos $Y$ y $Z$ en $H$, no necesitamos comprobar que $X$—$Z \notin E_a$ y $Y$—$Z \notin E_a$, porque, como $H$ es consistente con las restricciones, estas condiciones son verdad. ∎

### 3.8.3. Inicialización del algoritmo PC.

Como pasaba en el caso de los métodos basados en métrica+búsqueda, también tenemos que considerar el paso de inicialización del algoritmo. Un punto de inicio bastante común en los algoritmos basados en tests de independencia es un grafo completo y no dirigido $G_c = (\mathbf{X}, E_c)$, con $E_c = \{X$—$Y \mid X, Y \in \mathbf{X}\}$. En nuestro caso este grafo inicial debe ser modificado eliminando aquellas aristas que estén presentes en las restricciones de ausencia y orientando algunas aristas teniendo en cuenta las restricciones de existencia, ausencia y orden. Más concretamente, definamos los siguientes grafos:

- El grafo de las restricciones de ausencia no dirigidas:

  $G_{au} = (\mathbf{X}, E_{an})$, $E_{an} = \{X$—$Y \mid X$—$Y \in E_a\}$

- El grafo de las restricciones de ausencia invertidas:

  $G_{ai} = (\mathbf{X}, E_{ai})$, $E_{ai} = \{X \rightarrow Y \mid Y \rightarrow X \in E_a\}$

- La clausura transitiva del grafo de las restricciones de orden y las restricciones de existencia:

$G_{eot} = (\mathbf{X}, E_{ot})$, $E_{eot} = \{X \rightarrow Y \mid$ hay un camino dirigido de $X$ a $Y$ en $G_{re} \cup G_o\}$

El grafo completo y no dirigido $G_c$ debe ser reemplazado por el grafo $(G_c \setminus G_{au}) \cup G_{eot} \cup G_{ai}$. La figura 3.3 ilustra esta transformación.

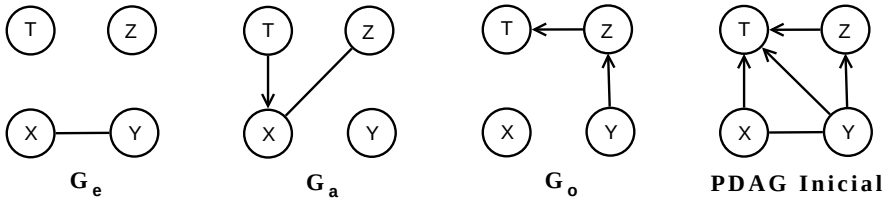


Figura 3.3: Transformando el grafo inicial, completo y no dirigido, en un PDAG consistente con las restricciones $G_e$, $G_a$ y $G_o$.

Sin embargo, puede aparecer un problema y es que el PDAG inicial no sea consistente con las restricciones. Por ejemplo, en el conjunto de las restricciones que aparecen en la figura 3.4 da lugar a un grafo $(G_c \setminus G_{au}) \cup G_{eot} \cup G_{ai}$ que no es consistente. Es bastante simple mostrar que esta situación sólo ocurrirá si el grafo $G_e \cup G_o \cup G_{ai}$ tiene un ciclo dirigido.

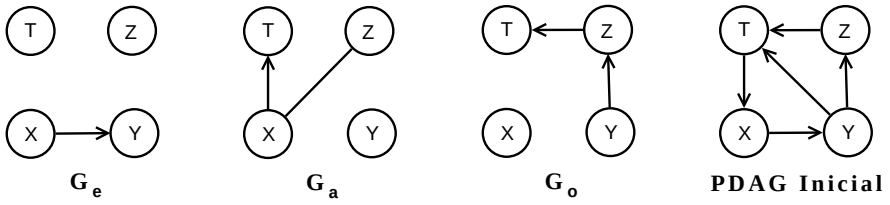


Figura 3.4: Transformando el grafo inicial, completo y no dirigido, en un grafo que no es consistente con las restricciones $G_e$, $G_a$ y $G_o$.

El motivo para esta situación es que en estos casos el conjunto de todas las restricciones *unidas*, implica otra restricción de ausencia que no ha sido indicada de manera explícita. Por ejemplo en la figura 3.4, las restricciones explícitas implican que el arco $T \rightarrow X$ no puede existir (de modo que la restricción $X \rightarrow T \in E_a$ de hecho es $X$—$T \in E_a$).

Una posible solución es detectar la situación y entonces eliminar el arco correspondiente (el arco $T \rightarrow X$ en el ejemplo). Sin embargo, hay casos donde más de un arco podría ser eliminado (es decir, el conjunto de restricciones

implícitas de ausencia que pueden ser deducidas a partir de las restricciones explícitas no forma una conjunción). Por ejemplo, el conjunto de restricciones $X \to Y \in E_e$, $Y \to Z \in E_o$ y $X \to T, T \to Z \in E_a$ implica $X—T \in E_a$ o $T—Z \in E_a$.

En estos casos tendríamos que elegir eliminar unos de estos arcos sin usar información adicional. Por esta razón creemos que una mejor solución es posponer la eliminación de estos arcos, permitiéndolos de forma temporal pero, en este caso, debemos tener en cuenta que los grafos intermedios que se obtengan no serán consistentes con las restricciones.

Una vez que la fase de eliminación de arcos y aristas guiada por los tests de independencia ha terminado, debemos eliminar los arcos adicionales (si los hubiera) necesarios para restaurar la consistencia. Esto se puede hacer de forma fácil mediante la comprobación de la presencia de ciclos en el grafo $H \cup G_o$, siendo $H$ el grafo actual, y posteriormente eliminar uno de los arcos en el ciclo que venga de una restricción de ausencia.

La ventaja de esta estrategia es que algunos arcos que pueden generar una falta de consistencia pueden haber sido eliminados anteriormente mediante los tests de independencia, y así nos evitamos realizar una selección arbitraria. En el último ejemplo, si el arco $Z \to T$ es eliminado por un test de independencia, nos evitamos el riesgo de tener que eliminar de forma arbitraria el arco $T \to X$ al principio (para después eliminar también el arco $Z \to T$).

### 3.8.4. Algoritmo PC con restricciones.

La adaptación propuesta del algoritmo PC para usar restricciones es, por lo tanto, bastante simple. Empezamos con el PDAG $(G_c \setminus G_{au}) \cup G_{eot} \cup G_{ai}$.

Posteriormente, la fase de eliminación de aristas del algoritmo PC es llevada a cabo, pero usando las condiciones (a) y (b) de la proposición 3.8.1 para la eliminación de arcos y aristas, respectivamente.

Después comprobamos la consistencia del PDAG resultante y eliminamos algunos arcos si es necesario, como hemos explicado previamente.

El siguiente paso en el algoritmo PC es la fase de detección de los patrones cabeza-cabeza, usando en este caso las condiciones (c) de la proposición 3.8.1.

Finalmente, la fase de orientación de las aristas restantes usando las reglas de coherencia, se lleva a cabo usando las condiciones (d) de 3.8.1. Si el grafo final $H$ no es un DAG, podemos direccionar las aristas restantes en cualquier

dirección que evite la creación de patrones cabeza-cabeza, y no creemos ciclos dirigidos en $H \cup G_o$.

## 3.9. Resultados experimentales.

En esta sección vamos a describir los experimentos que se han llevado a cabo para probar la utilización de restricciones en algoritmos de aprendizaje de redes bayesianas, y los resultados obtenidos.

El algoritmo de aprendizaje basado en el enfoque métrica+búsqueda considerado es la búsqueda local clásica, antes mencionado (con operadores de inserción, eliminación e inversión de arcos), usando la métrica bayesiana BDeu [150], donde el parámetro que representa el tamaño muestral equivalente es igual a 1 y se utiliza una estructura uniforme *a priori*. El algoritmo basado en tests de independencia usado es el PC.

Se han elegido cuatro problemas diferentes. La red *Alarm* (figura 3.6) que muestra las variables y relaciones relevantes para el sistema *Alarm Monitoring System* [29], una aplicación de diagnóstico para control de pacientes. Esta red contiene 37 variables y 46 arcos. *Insurance* [39] es una red para evaluar los riesgos de seguros de coches. La red *Insurance* (Figura 3.7) contiene 27 nodos y 52 enlaces. *Hailfinder* [3] es un sistema de pronóstico de granizo severo en el nordeste de Colorado. La red *Hailfinder* (figura 3.8) contiene 56 variables y 66 arcos. *Asia* (en la figura 3.5) es una pequeña red bayesiana que calcula la probabilidad de que un paciente tenga tuberculosis, cáncer de pulmón o bronquitis respectivamente, basándose en diferentes factores. Todas estas redes se han usado ampliamente en la literatura especializada para la comparación de algoritmos de aprendizaje estructural de redes bayesianas.

Las medidas usadas para estudiar el comportamiento de los métodos son: (1) El valor de la métrica (BDeu) para la red obtenida; esta medida es interesante debido a que es el criterio que guía la búsqueda local. (2) Tres medidas de diferencia estructural entre la red aprendida y la red original, que miden la capacidad de reconstruir la estructura del grafo: número de arcos añadidos (A), número de arcos eliminados (D) y el número de arcos invertidos (I) en la red aprendida con respecto a la red original. Para descartar diferencias o semejanzas ficticias entre dos redes, causadas por diferentes pero equivalen-

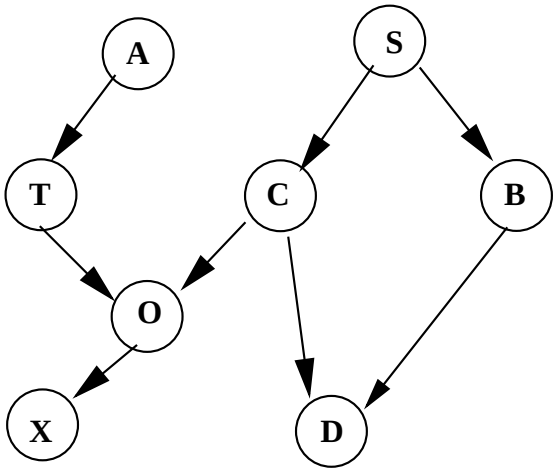


Figura 3.5: Red *Asia*.

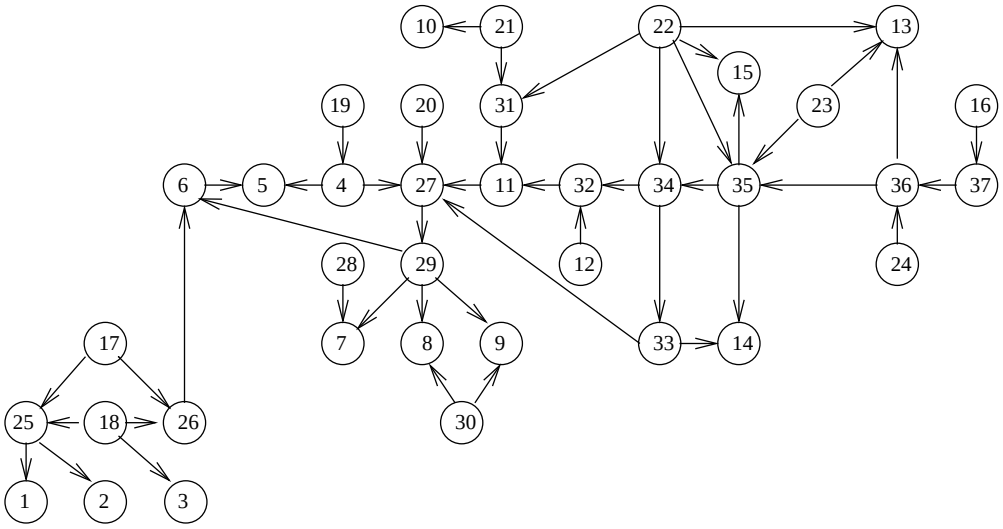


Figura 3.6: Red *Alarm*.

tes estructuras subDAG, antes de comparar, las dos redes se convierten a su correspondiente representación PDAG completo (también denominados grafos esenciales)[8], usando el algoritmo propuesto en [72]. (3) Una medida de la

[8]Un PDAG completo es un grafo acíclico parcialmente dirigido que es la representación

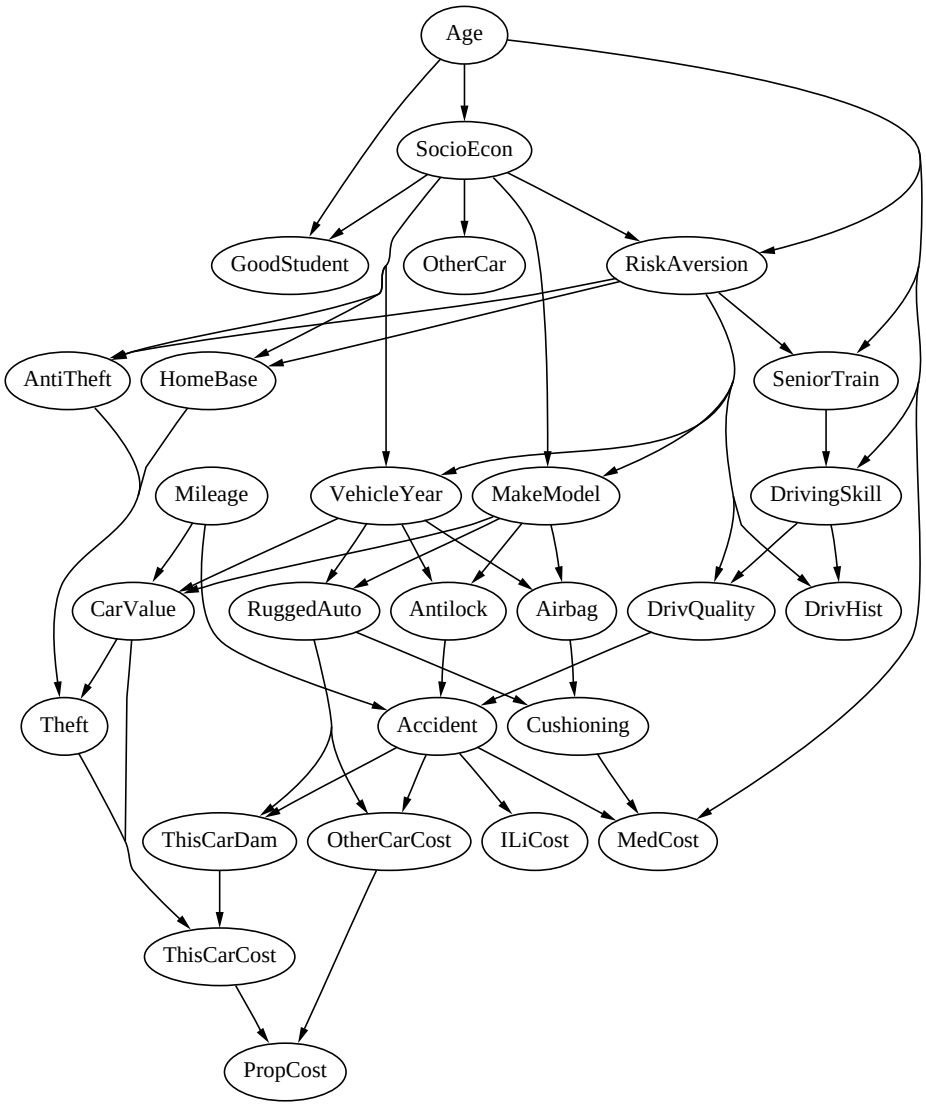


Figura 3.7: Red *Insurance*.

habilidad para reconstruir la distribución de probabilidad conjunta: vamos a usar la divergencia de Kullback-Leibler (KL) entre las distribuciones de probabilidad asociadas a la red original y a la red aprendida.

canónica de todos los DAGs que pertenecen a la misma clase de equivalencia.

Para cada conjunto de datos se han seleccionado de forma aleatoria unos porcentajes de restricciones de cada tipo, extraídas del conjunto de todas las posibles restricciones correspondiente a la red original. En concreto, si $G = (\mathbf{X}, E_G)$ es la red original, entonces cada arco $X \rightarrow Y \in E_G$ es una posible restricción de existencia (se puede seleccionar la restricción $X \rightarrow Y \in E_e$ si el arco está también presente en la representación del PDAG completo de $G$; en otro caso podemos usar la restricción $X—Y \in E_e$); cada arco $X \rightarrow Y \notin E_G$ es una posible restricción de ausencia (en caso de que también $Y \rightarrow X \notin E_G$ seleccionamos de manera aleatoria si usar la restricción $X \rightarrow Y \in E_a$ o $X—Y \in E_a$); finalmente, si hay un camino dirigido de $X$ a $Y$ en el PDAG completo de $G$ entonces $X \rightarrow Y \in E_o$ es una posible restricción de orden. Los porcentajes de restricciones usados han sido 10 %, 20 %, 30 % y 40 %. Se han ejecutado los algoritmos de aprendizaje para cada porcentaje de restricciones de cada tipo por separado, y también usando los tres tipos de restricciones a la vez.

Con cada red se han generado 10 bases de datos usando muestreo lógico, cada una de las cuales contenía 1000 instancias, excepto para *Asia*, cuyo tamaño muestral es 100. Los resultados que se muestran en los apartados siguientes, representan los valores medios de las distintas medidas tomadas a lo largo de 50 iteraciones (5 subconjuntos aleatorios para cada uno de los 10 conjuntos de datos).

### 3.9.1. Resultados para el algoritmo de búsqueda local

Las tablas 3.1–3.4 muestran los resultados obtenidos usando el algoritmo de búsqueda local. Incluyen los resultados obtenidos por el algoritmo de aprendizaje cuando no se usan restricciones (0 %). También muestran los valores de la divergencia KL de las redes originales, con los parámetros reaprendidos de las bases de datos correspondientes, lo cual nos sirve para comparar con las redes que sí usan restricciones. Las tablas 3.5 y 3.6 muestran los valores de la métrica BDeu de las distintas redes aprendidas, así como de las redes originales.

Primero analicemos los resultados desde el punto de vista de las diferencias estructurales. Se esperaba que el número de arcos eliminados, añadidos e invertidos decrecieran a la par que el número de restricciones de existencia, ausencia y orden, respectivamente, crecía. Este comportamiento es, de hecho, el que se ha observado claramente en los resultados. Además, otro

| | $G_e$, $G_a$, $G_o$ | | | | solo $G_e$ | | | | solo $G_a$ | | | | solo $G_o$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| % | KL | A | D | I | KL | A | D | I | KL | A | D | I | KL | A | D | I |
| 10 % | 0,1385 | 2,3 | 1,8 | 0,8 | 0,1448 | 2,8 | 2,0 | 0,8 | 0,1418 | 2,4 | 2,1 | 0,5 | 0,1491 | 2,9 | 2,3 | 0,6 |
| 20 % | 0,1319 | 1,7 | 1,4 | 0,5 | 0,1438 | 2,6 | 1,6 | 1,0 | 0,1409 | 2,1 | 2,1 | 0,6 | 0,1481 | 2,9 | 2,3 | 0,6 |
| 30 % | 0,1225 | 1,5 | 1,1 | 0,4 | 0,1378 | 2,5 | 1,4 | 0,9 | 0,1408 | 1,9 | 2,0 | 0,6 | 0,1481 | 2,9 | 2,3 | 0,6 |
| 40 % | 0,1169 | 1,1 | 0,8 | 0,3 | 0,1308 | 2,4 | 1,1 | 0,9 | 0,1376 | 1,4 | 1,8 | 0,5 | 0,1485 | 2,9 | 2,3 | 0,6 |
| 0 % | 0,1491 | 2,9 | 2,3 | 0,6 | KL red *Asia* original (reaprendida): 0,0955 | | | | | | | | | | | |

Tabla 3.1: Resultados medios obtenidos para *Asia* usando búsqueda local.

| | $G_e$, $G_a$, $G_o$ | | | | solo $G_e$ | | | | solo $G_a$ | | | | solo $G_o$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| % | KL | A | D | I | KL | A | D | I | KL | A | D | I | KL | A | D | I |
| 10 % | 0,2875 | 6,7 | 3,0 | 7,9 | 0,2896 | 8,0 | 3,0 | 12,5 | 0,3419 | 9,9 | 3,9 | 15,7 | 0,3163 | 9,5 | 3,6 | 13,7 |
| 20 % | 0,2620 | 4,2 | 2,4 | 4,3 | 0,2595 | 6,5 | 2,4 | 9,5 | 0,3608 | 8,7 | 4,2 | 12,8 | 0,3079 | 8,9 | 3,6 | 11,5 |
| 30 % | 0,2338 | 2,7 | 2,0 | 2,7 | 0,2435 | 5,7 | 2,0 | 7,1 | 0,3401 | 6,5 | 4,1 | 9,6 | 0,2833 | 7,6 | 3,4 | 8,5 |
| 40 % | 0,2240 | 2,1 | 1,7 | 1,6 | 0,2309 | 4,8 | 1,6 | 5,4 | 0,3183 | 4,9 | 3,6 | 7,0 | 0,2714 | 7,0 | 3,3 | 6,7 |
| 0 % | 0,3199 | 10,5 | 3,7 | 18,4 | KL red *Alarm* original (reaprendida): 0,2112 | | | | | | | | | | | |

Tabla 3.2: Resultados medios obtenidos para *Alarm* usando búsqueda local.

| | $G_e$, $G_a$, $G_o$ | | | | solo $G_e$ | | | | solo $G_a$ | | | | solo $G_o$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| % | KL | A | D | I | KL | A | D | I | KL | A | D | I | KL | A | D | I |
| 10 % | 0,5592 | 3,3 | 15,3 | 11,5 | 0,5871 | 4,6 | 15,9 | 13,6 | 0,6053 | 4,7 | 17,5 | 14,3 | 0,5984 | 5,4 | 17,9 | 14,2 |
| 20 % | 0,5403 | 2,1 | 13,4 | 8,9 | 0,5571 | 3,5 | 13,9 | 12,3 | 0,5819 | 3,6 | 17,0 | 13,0 | 0,5729 | 4,9 | 17,8 | 13,0 |
| 30 % | 0,5144 | 1,0 | 11,3 | 6,3 | 0,5293 | 2,5 | 11,8 | 9,8 | 0,5685 | 2,7 | 16,6 | 11,1 | 0,5614 | 4,5 | 17,6 | 11,8 |
| 40 % | 0,5092 | 0,5 | 9,3 | 4,8 | 0,5263 | 1,9 | 9,8 | 7,3 | 0,5587 | 2,0 | 16,2 | 8,1 | 0,5513 | 4,1 | 17,4 | 10,4 |
| 0 % | 0,6295 | 6,1 | 18,3 | 16,1 | KL red *Insurance* original (reaprendida): 0,5531 | | | | | | | | | | | |

Tabla 3.3: Resultados medios obtenidos para *Insurance* usando búsqueda local.

| | $G_e$, $G_a$, $G_o$ | | | | solo $G_e$ | | | | solo $G_a$ | | | | solo $G_o$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| % | KL | A | D | I | KL | A | D | I | KL | A | D | I | KL | A | D | I |
| 10 % | 1,4409 | 11,7 | 16,2 | 13,2 | 1,4322 | 14,1 | 16,6 | 16,4 | 1,4329 | 13,4 | 17,7 | 15,6 | 1,4382 | 15,0 | 18,3 | 14,8 |
| 20 % | 1,4974 | 9,2 | 14,3 | 12,4 | 1,4946 | 13,3 | 14,9 | 16,7 | 1,4249 | 11,3 | 17,3 | 13,1 | 1,4536 | 15,1 | 18,7 | 13,6 |
| 30 % | 1,5209 | 6,7 | 12,3 | 10,4 | 1,5330 | 11,9 | 13,0 | 15,7 | 1,4049 | 8,9 | 16,2 | 11,5 | 1,4465 | 14,7 | 18,6 | 12,0 |
| 40 % | 1,5453 | 4,8 | 10,7 | 8,6 | 1,5415 | 11,1 | 11,2 | 14,4 | 1,3886 | 7,2 | 15,6 | 9,8 | 1,4517 | 14,5 | 18,7 | 10,6 |
| 0 % | 1,4216 | 15,5 | 18,1 | 17,3 | KL red *Hailfinder* original (reaprendida): 1,1609 | | | | | | | | | | | |

Tabla 3.4: Resultados medios obtenidos para *Hailfinder* usando búsqueda local.

efecto menos obvio, observado frecuentemente en los experimentos, es que el uso de cualquiera de los tres tipos de restricciones tiende también a disminuir las otras medidas de diferencia estructural. Por ejemplo, la restricciones de existencia disminuyen el número de arcos borrados, pero también el número de arcos añadidos o invertidos.

Con respecto al análisis de los resultados desde el punto de vista de la divergencia KL, tenemos que distinguir *Hailfinder* de los otros tres conjuntos de datos. En estos, el uso de cada tipo de restricción conduce a mejores estructuras de red, y las mejoras se incrementan casi sistemáticamente con el número de restricciones usadas. Sin embargo, hay unos pocos casos (con *Alarm*) donde el uso de restricciones de ausencia da lugar a peores resultados que en la búsqueda local sin restricciones. Creemos que la explicación de este comportamiento radica en el siguiente hecho: cuando un algoritmo de búsqueda local se equivoca en la dirección de algún arco que conecta dos nodos[9], entonces el algoritmo tiende a 'cruzar' los padres de dichos nodos para compensar la orientación errónea; si alguno de estos arcos 'cruzados' son usados como restricciones de ausencia, entonces el algoritmo no puede compensar el error cometido y acaba terminando en una configuración peor. Estos resultados sugieren otra forma de usar las restricciones de ausencia: una vez que el algoritmo ha encontrado un máximo local, usando sólo restricciones de existencia y de orden, podemos eliminar aquellos arcos prohibidos e iniciar otra búsqueda local.

No obstante, el caso de *Hailfinder* es completamente diferente, los tres tipos de restricciones dan lugar a peores redes, cuantas más restricciones usamos mayor es la divergencia de KL (excepto para el caso de las restricciones de ausencia). Por el momento, no tenemos una explicación para este comportamiento inesperado de la divergencia KL para el conjunto de datos *Hailfinder*.

| | Asia | | | | Alarm | | | |
|---|---|---|---|---|---|---|---|---|
| | $G_e, G_a, G_o$ | solo $G_e$ | solo $G_a$ | solo $G_o$ | $G_e, G_a, G_o$ | solo $G_e$ | solo $G_a$ | solo $G_o$ |
| 10 % | -218,97 | -218,68 | -218,66 | -218,35 | -11184,80 | -11192,55 | -11253,17 | -11233,99 |
| 20 % | -219,50 | -219,15 | -218,73 | -218,38 | -11151,20 | -11157,59 | -11264,57 | -11226,59 |
| 30 % | -219,67 | -219,22 | -219,01 | -218,38 | -11123,20 | -11134,51 | -11237,61 | -11190,76 |
| 40 % | -220,00 | -219,41 | -219,25 | -218,38 | -11113,99 | -11119,93 | -11209,80 | -11171,16 |
| 0 % | -218,35 | BDeu red original: -258,91 | | | -11233,37 | BDeu red original: -11256,62 | | |

Tabla 3.5: Valores medios de la métrica BDeu para *Asia* y *Alarm*.

[9]Esta situación puede ser bastante frecuente en las primeras etapas del proceso de búsqueda.

| | Insurance | | | | Hailfinder | | | |
|---|---|---|---|---|---|---|---|---|
| | $G_e, G_a, G_o$ | solo $G_e$ | solo $G_a$ | solo $G_o$ | $G_e, G_a, G_o$ | solo $G_e$ | solo $G_a$ | solo $G_o$ |
| 10 % | -14105,75 | -14137,72 | -14115,24 | -14113,08 | -52638,00 | -52612,30 | -52599,14 | -52595,80 |
| 20 % | -14100,71 | -14129,56 | -14082,38 | -14080,21 | -52785,04 | -52720,74 | -52616,60 | -52607,93 |
| 30 % | -14117,11 | -14138,98 | -14060,97 | -14068,16 | -52977,60 | -52868,29 | -52620,29 | -52608,47 |
| 40 % | -14158,52 | -14178,02 | -14047,71 | -14052,52 | -53207,97 | -53020,67 | -52641,89 | -52622,05 |
| 0 % | -14152,92 | BDeu true network: -14439,10 | | | -52580,32 | BDeu true network: -55268,49 | | |

Tabla 3.6: Valores medios de la métrica BDeu para *Insurance* y *Hailfinder*.

Finalmente, en lo que respecta a los valores de la métrica BDeu, se puede observar que las redes aprendidas, siempre tienen valores mayores de la métrica BDeu (y, por tanto, mejores) que las redes originales, lo cual indica cierto tipo de sobreajuste a los datos. Además, conforme se incrementan el número de restricciones, los valores de la métrica BDeu tienden a decrementarse para *Asia* y *Hailfinder*, y tienden a crecer para *Alarm* e *Insurance*. Creemos que esto es debido al alto grado de sobreajuste de la métrica BDeu en los dos primeros casos, debido a tamaño muestral relativamente pequeño.

### 3.9.2. Resultados para el algoritmo PC

En algunos experimentos preliminares hemos observado pobres resultados cuando usábamos restricciones de ausencia, especialmente en el caso de la divergencia KL y el número de arcos invertidos. Creemos que el motivo para este comportamiento es el siguiente: el algoritmo no realiza tests de independencia para cualquier par de nodos $X$ e $Y$ pertenecientes a una restricción de ausencia no dirigida. Por lo tanto, si estos nodos se relacionan a través de otro nodo $Z$, $X$—$Z$—$Y$, no tenemos la información necesaria para determinar si forman un patrón cabeza-cabeza (sabiendo que $Z$ no está en el subconjunto que separa a $X$ e $Y$). Por este motivo, trabajamos con una versión modificada del algoritmo de tal modo que no se eliminan del grafo inicial los enlaces $X$—$Y \in G_a$. Son eliminados después del paso en el que el algoritmo elimina enlaces usando tests de independencia. Los resultados obtenidos con esta modificación se muestran en las tablas 3.7–3.10.

En este caso, el uso de restricciones siempre conduce a mejores estructuras de red que el algoritmo PC sin restricciones, desde el punto de vista de la divergencia KL. Con respecto a las diferencias estructurales, todos los tipos de restricciones reducen el número de arcos borrados (posiblemente esto es debido a que se llevan a cabo menos tests de independencia). No obstante,

| | $G_e$, $G_a$, $G_o$ | | | | solo $G_e$ | | | | solo $G_a$ | | | | solo $G_o$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| % | KL | A | D | I | KL | A | D | I | KL | A | D | I | KL | A | D | I |
| 10 % | 0,2520 | 0,3 | 4,0 | 2,7 | 0,2639 | 0,2 | 4,1 | 3,0 | 0,2839 | 0,2 | 4,8 | 2,8 | 0,2847 | 0,2 | 4,8 | 2,9 |
| 20 % | 0,2224 | 0,4 | 3,3 | 2,0 | 0,2388 | 0,2 | 3,4 | 2,8 | 0,2796 | 0,2 | 4,8 | 2,7 | 0,2802 | 0,3 | 4,7 | 2,8 |
| 30 % | 0,2023 | 0,5 | 2,6 | 1,6 | 0,2242 | 0,3 | 3,0 | 2,6 | 0,2837 | 0,4 | 4,7 | 2,5 | 0,2740 | 0,4 | 4,7 | 2,6 |
| 40 % | 0,1818 | 0,5 | 2,1 | 1,1 | 0,2145 | 0,4 | 2,4 | 2,4 | 0,2774 | 0,4 | 4,6 | 2,2 | 0,2735 | 0,6 | 4,7 | 2,3 |
| 0 % | 0,2895 | 0,2 | 4,8 | 2,8 | KL red *Asia* original (reaprendida): 0,0955 | | | | | | | | | | | |

Tabla 3.7: Resultados medios obtenidos para *Asia* usando PC.

| | $G_e$, $G_a$, $G_o$ | | | | solo $G_e$ | | | | solo $G_a$ | | | | solo $G_o$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| % | KL | A | D | I | KL | A | D | I | KL | A | D | I | KL | A | D | I |
| 10 % | 2,2590 | 1,5 | 13,5 | 7,6 | 2,3850 | 1,7 | 15,0 | 8,5 | 2,7508 | 1,5 | 16,9 | 9,2 | 2,7265 | 1,6 | 17,1 | 8,8 |
| 20 % | 1,5953 | 1,2 | 9,5 | 6,4 | 1,9995 | 1,8 | 12,6 | 7,2 | 2,5969 | 1,5 | 15,8 | 8,6 | 2,6471 | 1,7 | 16,5 | 8,3 |
| 30 % | 1,1720 | 1,1 | 6,8 | 4,5 | 1,7976 | 1,9 | 10,5 | 6,2 | 2,3655 | 1,2 | 14,0 | 7,7 | 2,4914 | 1,6 | 15,2 | 8,3 |
| 40 % | 0,8204 | 1,0 | 4,9 | 2,9 | 1,5681 | 1,9 | 8,5 | 5,6 | 2,0665 | 1,1 | 12,1 | 6,9 | 2,3195 | 1,7 | 14,2 | 8,2 |
| 0 % | 2,7482 | 1,7 | 17,8 | 9,6 | KL red *Alarm* original (reaprendida): 0,2112 | | | | | | | | | | | |

Tabla 3.8: Resultados medios obtenidos para *Alarm* usando PC.

| | $G_e$, $G_a$, $G_o$ | | | | solo $G_e$ | | | | solo $G_a$ | | | | solo $G_o$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| % | KL | A | D | I | KL | A | D | I | KL | A | D | I | KL | A | D | I |
| 10 % | 2,2192 | 1,1 | 26,9 | 11,4 | 2,4062 | 1,4 | 27,8 | 12,0 | 2,3718 | 1,1 | 31,2 | 7,9 | 2,3385 | 1,4 | 31,3 | 7,5 |
| 20 % | 1,8538 | 1,0 | 22,7 | 13,0 | 2,2920 | 1,4 | 24,7 | 14,4 | 2,2248 | 1,1 | 30,4 | 8,4 | 2,2134 | 1,4 | 30,7 | 7,9 |
| 30 % | 1,6081 | 0,8 | 18,1 | 13,5 | 2,1726 | 1,3 | 20,8 | 15,9 | 2,0282 | 1,1 | 29,4 | 9,1 | 2,1007 | 1,3 | 30,2 | 7,8 |
| 40 % | 1,4259 | 0,5 | 14,8 | 13,4 | 2,0900 | 1,3 | 17,7 | 17,2 | 1,8690 | 0,6 | 28,3 | 9,6 | 1,9995 | 1,3 | 29,7 | 7,9 |
| 0 % | 2,4314 | 1,4 | 31,6 | 7,4 | KL red *Insurance* original (reaprendida): 0,5531 | | | | | | | | | | | |

Tabla 3.9: Resultados medios obtenidos para *Insurance* usando PC.

| | $G_e$, $G_a$, $G_o$ | | | | solo $G_e$ | | | | solo $G_a$ | | | | solo $G_o$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| % | KL | A | D | I | KL | A | D | I | KL | A | D | I | KL | A | D | I |
| 10 % | 8,2310 | 12,2 | 32,3 | 9,7 | 8,4367 | 11,1 | 32,8 | 10,4 | 8,9021 | 11,8 | 36,3 | 8,8 | 9,0700 | 10,9 | 36,6 | 9,0 |
| 20 % | 7,7158 | 13,0 | 27,9 | 9,9 | 8,2728 | 11,3 | 28,8 | 11,4 | 8,6166 | 12,3 | 35,9 | 8,3 | 8,9622 | 11,2 | 36,3 | 8,8 |
| 30 % | 7,4199 | 14,0 | 23,6 | 10,1 | 8,2550 | 11,5 | 24,9 | 12,6 | 8,2566 | 13,6 | 35,1 | 7,7 | 8,8277 | 11,6 | 35,9 | 8,3 |
| 40 % | 7,4649 | 15,7 | 19,4 | 10,3 | 8,3642 | 11,7 | 20,9 | 13,5 | 8,0320 | 15,4 | 34,3 | 7,2 | 8,7295 | 11,7 | 35,5 | 7,8 |
| 0 % | 9,1548 | 10,7 | 36,8 | 9,4 | KL red *Hailfinder* original (reaprendida): 1,1609 | | | | | | | | | | | |

Tabla 3.10: Resultados medios obtenidos para *Hailfinder* usando PC.

por el mismo motivo, el número de arcos añadidos tiende a incrementarse (excepto en el caso de *Insurance*). El número de arcos invertidos tiende a disminuir (excepto, otra vez, en el caso de *Insurance*).

## 3.10. Discusión.

Se han definido formalmente tres tipos de restricciones estructurales para redes bayesianas, que hemos denominado restricciones de existencia, ausencia y orden. Las restricciones presentadas nos van a permitir incorporar conocimiento de forma visualmente interpretable, siendo así una herramienta intuitiva en su utilización por un experto.

En efecto, si dos variables están relacionadas habrá una arista que las una, si además hay una relación de causalidad esa arista tendrá una dirección y será por tanto un arco que indique la dirección de la causalidad (restricciones de existencia).

En cambio, si dos variables no están relacionadas no existirá un enlace entre ellas o, si sabemos, que una variable no es causa de otra, podremos representarlo prohibiendo expresamente un arco de la primera a la segunda variable (restricciones de ausencia).

Las relaciones de orden reflejan precedencia temporal o funcional entre variables, por ejemplo, la expresión de un gen puede provocar la expresión de otro pero el proceso no tiene por qué ser inmediato, pues ambas expresiones pueden estar en un mismo camino de regulación genética en el ciclo celular: de esta forma sabemos que el primer gen tendrá un orden mayor al segundo.

En la aplicación práctica de las restricciones, se ha estudiado su uso en algoritmos de aprendizaje de redes bayesianas basados en métrica+búsqueda y en tests de independencia. Se ha ilustrado para el caso concreto de un algoritmo de búsqueda local y para el algoritmo PC.

Los resultados experimentales muestran que el uso de información adicional en forma de restricciones mejoran las estructuras de las redes obtenidas.

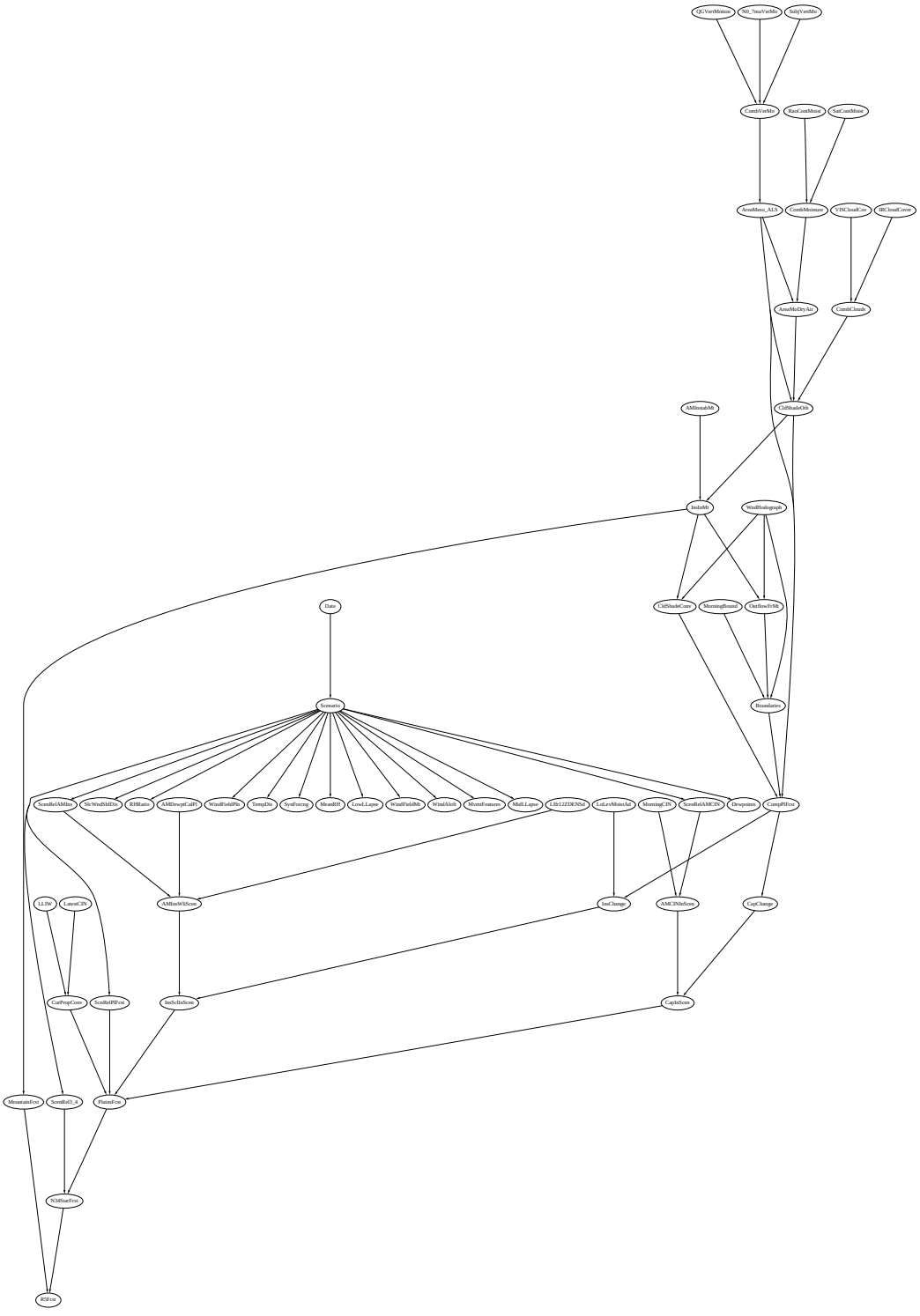




Figura 3.8: Red *Hailfinder*.

# Capítulo 4

# Árboles de clasificación usando una estimación bayesiana.

Los *árboles de decisión* son un tipo de clasificadores supervisados que se basan en el particionamiento recursivo del espacio de valores de los atributos del problema. Se dividen en *árboles de clasificación* (o *clasificadores jerárquicos*) cuando la clase es una variable discreta o *árboles de regresión* cuando la salida es una variable continua. Nosotros nos centraremos en el primer caso, en los árboles de clasificación.

El objetivo es ir dividiendo el conjunto de casos en base a un criterio y usando una única variable en cada partición, hasta que al final, idealmente, en cada una de las distintas particiones realizadas no haya más que casos pertenecientes a una misma clase.

La representación del conocimiento adquirido en un árbol de clasificación es relativamente simple. Se puede interpretar como un conjunto de reglas compactadas en forma de árbol, dónde cada nodo no terminal se etiqueta con una variable atributo en la que se realiza un test que produce ramificaciones correspondientes a sus posibles valores. Los nodos hoja se etiquetan con un valor de la variable a clasificar, como se puede ver en la figura 4.1 o con las probabilidades asociadas a cada clase 4.2.

Los árboles de clasificación se han aplicado con éxito en distintos campos como son: astronomía [245, 298], biología [313], medicina [202, 108, 199, 229, 177, 362, 357, 113], selección de variables [1] o en física [52].

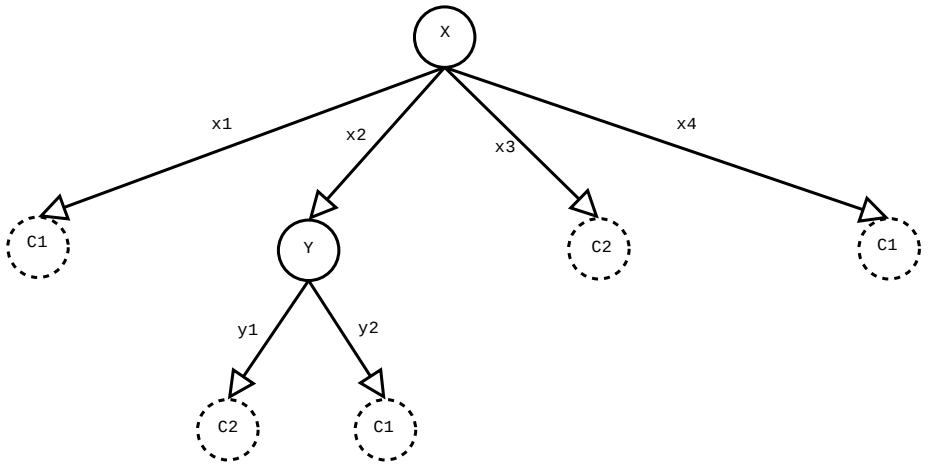


Figura 4.1: Árbol de clasificación.

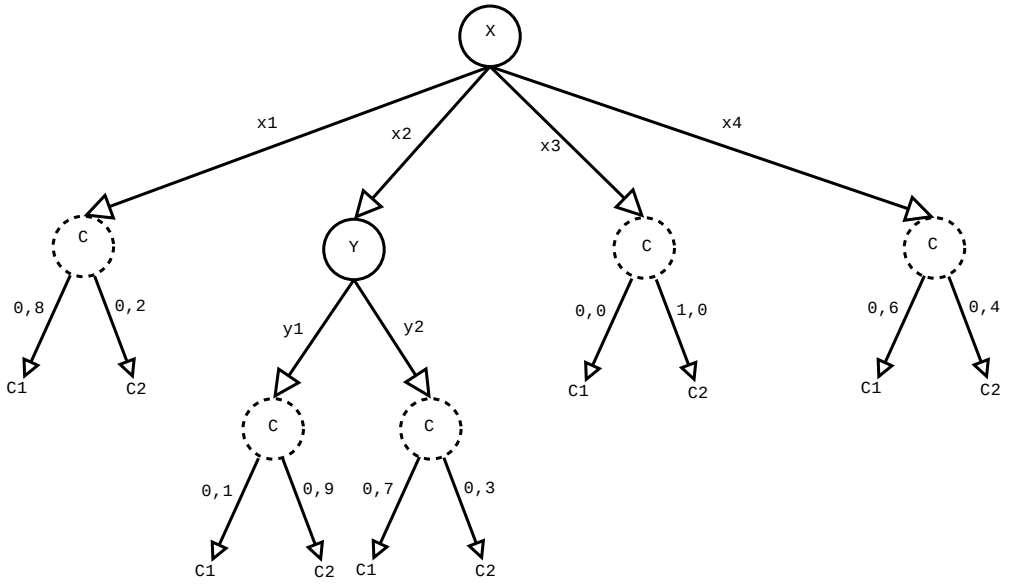


Figura 4.2: Árbol de clasificación con probabilidades en las hojas.

## 4.1. Árboles de clasificación.

Los árboles de clasificación tienen su origen en el trabajo de Hunt [160], aunque fue con la aparición del algoritmo ID3 de Quinlan en 1979 [277],

cuando cobraron importancia; posteriormente Quinlan presentó el algoritmo C4.5 [279] que es una mejora del anterior y que obtiene mejores resultados.

En los algoritmos propuestos por Quinlan, ID3 y C4.5, la creación del árbol de clasificación se basa en los siguientes puntos:

- Determinación del procedimiento para elegir un nodo raíz del árbol actual (se empieza con el árbol vacío).
- Determinación del procedimiento para ramificar, así como el criterio de parada para determinar que estamos en un nodo hoja.
- Determinación del criterio de selección de la variable a clasificar que se introduce en un nodo hoja, o de las probabilidades para cada clase.
- Determinación del procedimiento de refinamiento (poda).

Estos dos métodos comienzan con el árbol vacío y en cada nivel seleccionan la variable con mayor ganancia de información (o con una mayor razón de ganancia, en el caso de C4.5), con respecto a la variable a clasificar. Una vez seleccionada la variable se ramifica por cada uno de sus valores. Se repite el proceso para cada rama, pero utilizando un subconjunto de casos. Si estamos repitiendo el proceso para la rama en la que $X = x_j$, utilizaremos el subconjunto de casos resultante de quedarnos con aquellos casos en los que se verifica que $X = x_j$, es decir, el subconjunto de casos que para la variable padre toma el valor al de la rama en la que estamos. Para cada rama, se expande el árbol hasta que se den las condiciones necesarias para dejar de hacerlo (demasiada profundidad, todos los casos de la variable a clasificar pertenecen a la misma clase, etc.). Posteriormente para evitar el sobreajuste del árbol al conjunto de datos, se realiza una poda para que el árbol generalice mejor.

Además de estos dos algoritmos, vamos a presentar una nueva aproximación que guarda muchas similitudes con ID3 y C4.5, siendo más sencillo. Para escoger la variable a clasificar utiliza, básicamente, sólo el concepto de entropía, cuyas probabilidades se han calculado usando una distribución *a priori* de Dirichlet. La única complejidad que introduce es que se utiliza una condición de parada más.

La utilización de la distribución de Dirichlet para la construcción de árboles de clasificación es nueva, aunque otros autores [2] han utilizado ya el modelo impreciso de Dirichlet [347] consiguiendo buenos resultados, con un proceso más complejo, que el que se va a describir aquí. También se ha utilizado la distribución de Dirichlet en otros métodos de aprendizaje como en redes neuronales [46] o en redes bayesianas [266].

Cuando en los nodos hoja se utilizan probabilidades para cada clase, algunos autores han utilizado la corrección de Laplace para suavizar dichas probabilidades [276], en muestras pequeñas. La corrección de Laplace se puede ver como una estimación bayesiana de los parámetros esperados de una distribución multinomial usando una distribución *a priori* de Dirichlet. La diferencia de nuestro enfoque es que nosotros no sólo las usamos en la estimación de las probabilidades de las hojas, sino también en la estimación de las probabilidades que se usan en la expresión de la ganancia de información, lo que hace que este criterio mejore su comportamiento.

### 4.1.1. Algoritmos ID3 y C4.5

Vamos a presentar con más profundidad los algoritmos ID3 y C4.5 para la construcción de árboles de clasificación. Aprovecharemos que el segundo es una evolución del primero, para ir presentándolos a la vez, remarcando las diferencias.

Supongamos que $C$ es la variable a clasificar. En el proceso de construcción del árbol cada nodo interior se etiqueta con una variable y cada arco que sale de él con uno de sus valores. Cada nodo tiene asociado un subconjunto $D$ de ejemplos del conjunto de datos, así en el nodo raíz se tendrá el conjunto de datos entero. Si todos los ejemplos de $D$ tuviesen igual valor para la variable a clasificar, entonces ese nodo sería un nodo hoja cuya etiqueta sería el valor de la clase. En otro caso, los valores de la variable a clasificar son diferentes, por lo que mediante un *test*, determinamos la variable que mejor divide el conjunto $D$ respecto de $C$.

Una vez determinada la mejor variable, se generan tantos nodos hijos como valores distintos tenga esa variable. A cada nodo hijo se le asigna el subconjunto de $D$ de los ejemplos en los que la variable elegida tiene el valor correspondiente a ese hijo.

De esta forma, se sigue una estrategia *divide y vencerás* donde en cada paso se va dividiendo el conjunto de datos hasta alcanzar subconjuntos del mismo cuya variable a clasificar tenga el mismo valor o hasta que se cumpla alguna otra condición de parada.
La elección de la mejor variable para seguir expandiendo el árbol se basa en que se escogerá la variable que *más información* proporcione sobre la variable a clasificar, es decir, que a partir de esa variable se pueda predecir con más probabilidad la clase.

La formalización de *medida de la información* fue introducida por Shannon en 1948 [309]. A continuación expondremos estos conceptos para una mejor compresión de los métodos ID3 y C4.5.

Shannon definió la siguiente medida para la información basada en el concepto de probabilidad:

**Definición.** Sea $E$ un evento que pueda suceder con una probabilidad $p(E)$. Cuando $E$ tiene lugar, decimos que hemos recibido $I(E) = log_a(1/p(E))$ unidades de información. Si $a = 2$ la unidad de información recibe el nombre de *bit*, y si $p(E) = 1/2$ entonces $I(E) = 1$ bit; por lo que definimos el bit como la cantidad de información proporcionada por la especificación de uno de dos sucesos equiprobables.

Shannon generaliza el concepto de información para un mecanismo generador de información denominado fuente de información de memoria nula.

**Definición.** Sea $S = s_1, s_2, ...s_n$ un alfabeto finito y fijo que se utilizará para transmitir la información proporcionada por un dispositivo generador denominado fuente. Si los símbolos, o elementos de $S$, son trasmitidos de acuerdo a una distribución de probabilidad fija: $p(s1), p(s2), ...p(s_n)$, entonces se denomina fuente de información de memoria nula. Una fuente de información se denomina de memoria nula cuando la emisión de un símbolo es independiente de los símbolos emitidos anteriormente. Por abuso del lenguaje se suele identificar la fuente con el propio $S$.

Veamos ahora el concepto de entropía que es el que usaremos posteriormente.

**Definición.** Sea $S = s_1, s_2, ...s_n$ un alfabeto de una fuente de información de memoria nula. La cantidad media de información por símbolo del alfabeto $S$, se denomina entropía de $S$, se presenta por $H(S)$, y tiene la siguiente expresión:

$$H(S) = \sum_{i=1}^{n} p(s_i) \log_2 \frac{1}{p(s_i)}$$

La entropía también se puede interpretar usando el concepto, más intuitivo, de *incertidumbre*, es decir, puede verse $H(S)$ como el valor medio de la incertidumbre de un observador antes de conocer una salida de una fuente de información de memoria nula.

En ID3 y posteriormente en C4.5, Quinlan hace una representación intuitiva de los árboles de clasificación, donde cada nodo se puede ver como una fuente de información de memoria nula, cuyo alfabeto tiene tantos símbolos como clases. Cada nodo al tener almacenado un subconjunto de casos del conjunto de datos, guarda una información media sobre las clases de esos casos.

En ID3, el mejor atributo $X$ para ramificar el árbol dado el conjunto de datos $D$, es aquel que maximiza la ganancia de información. Para calcular la ganancia de cada atributo se realizan los siguientes pasos:

- Se calcula la entropía de la clase $C$ antes de ramificar, o lo que viene a ser lo mismo, la entropía de $C$ para el conjunto $D$:

  $$H_D(C) = \sum_{i=1}^{n} p_i log_2 \frac{1}{p_i}$$

  donde $p_i$ representa la probabilidad (calculada por frecuencias relativas) de la clase $i$ en $D$ y $n$ es el número de clases.

- Se calcula el valor medio de la entropía en el nivel siguiente, generado por el atributo $X$:

  $$H_D(C|X) = \sum_{j=1}^{k} p_D(X = x_j) H_{D_j}(C)$$

donde $x_1, x_2, ..x_k$ son los valores del atributo $X$, $p_D(X = x_j)$ representa la probabilidad condicionada de que $X = x_j$ en $D$ y $D_j$ es el subconjunto de $D$, $D_j \subset D$, tal que sus elementos tienen el valor $X = x_j$.

- Se denomina *ganancia de información* de la variables $X$, a la diferencia de entropía de la entropía de $C$ para el conjunto $D$ menos la entropía media del siguiente nivel si ramificásemos por $X$:

$$Ganancia_D(C|X) = H_D(C) - H_D(C|X)$$

En ID3, se escoge la variable con mayor ganancia de información.

En C4.5 además se calcula la *Información de ruptura* y la *razón de ganancia*. Para un conjuntos de datos $D$ y un atributo $X$, se define la razón de ganancia de la siguiente forma:

$$RazónGanancia(X|C) = \frac{Ganancia_D(C|X)}{InfoRuptura_D(X)}$$

donde,

$$InfoRuptura_D(X) = \sum_{j=1}^{k} p_D(X = x_j) \log_2 \frac{1}{p_D(X=x_j)}$$

.

El test basado en el criterio de máxima ganancia, tiende a escoger aquellas variables con un mayor número de valores. Esto es debido a que cuantas más particiones se hagan debido a los valores de la variable, la entropía de un nuevo nodo será, normalmente, menor. En C4.5 se utiliza la razón de ganancia para paliar esta tendencia.

El algoritmo C4.5 también se diferencia de ID3 porque admite datos continuos, dividiendo el dominio de la variable continua en subintervalos por el valor que tiene una mayor ganancia de información. Además C4.5 admite valores desconocidos en los ejemplos, modificando los criterios de ganancia e información de ruptura.

### 4.1.2. Condiciones de parada

Las condiciones de parada para dejar de ramificar un árbol de clasificación se utilizan cuando su crecimiento no parece mejorar la capacidad predictiva del árbol. Las reglas que se utilizan son las siguientes:

1. Todos los casos pertenecen a la misma clase.

2. Todos los casos tienen los mismos valores para las variables, aunque no necesariamente coincidan en los valores de la variable a clasificar.

3. Se ha llegado a un nivel de profundidad en la construcción del árbol superior a un límite.

4. No hay rechazo de independencia entre una variable X y la variable a clasificar en un test de hipótesis (por ejemplo, en un test chi cuadrado $\chi^2$).

## 4.2. Árboles de clasificación usando una distribución de Dirichlet.

En esta sección describimos una nueva aportación de esta memoria para construir árboles de clasificación. Sigue el mismo esquema de ID3 o C4.5, aunque con pequeñas diferencias. La idea principal es la siguiente: en los algoritmos clásicos cuando se estima la entropía se usan valores numéricos para las probabilidades que son estimadas mediante máxima verosimilitud.

Si queremos calcular la probabilidad $p_i$ de que una variable discreta aleatoria $Y$, sea igual a un valor $(Y = y_i)$ dado un conjunto de datos $D$. Calculamos la frecuencia absoluta $c_i$ de la ocurrencia de ese evento en el conjunto. Por tanto, la probabilidad estimada por máxima verosimilitud, usando frecuencias relativas, sería:

$$p_i = c_i/m$$

Donde $m$ es el número de casos del conjunto $D$.

Obsérvese que si el tamaño del conjunto es 1 y en el único caso que tenemos $Y$ toma el valor $y_i$, asumimos que la probabilidad de $(Y = y_i)$ es $p_i = 1$. Es bien sabido que este enfoque conlleva importantes problemas de sobreajuste.

Un procedimiento mucho más apropiado es estimar los valores de las probabilidades para una distribución multinomial usando un procedimiento bayesiano con una distribución *a priori* de Dirichlet [347, 129]. Nuestro método utiliza probabilidades estimadas con la distribución *a priori* de Dirichlet

para calcular la entropía y la entropía condicionada. Esto tiene consecuencias importantes. La primera es que la ganancia de información puede ser negativa. De esta forma obtenemos una nueva condición de parada: cuando una ramificación produzca una ganancia de información negativa paramos la expansión por esa rama del árbol. También nos proporciona una forma más conveniente en la selección de variables por las cuales ramificar. Finalmente, también obtenemos árboles más simples, los cuales pueden ser podados posteriormente, pero esto es más rápido ya que se aplica a árboles más pequeños.

La distribución de Dirichlet es apropiada para la distribución *a priori* de probabilidades multinomiales: $\theta = (\theta_1, \ldots, \theta_n)$ con $\sum_{i=1}^{n} \theta_i = 1$ para los diferentes valores $(y_1, \ldots, y_n)$ de una variable discreta aleatoria $Y$. Depende de un par de parámetros $(s, t)$, donde $t = (t_1, t_2, \ldots, t_n)$ con $t_i \geq 0, s \geq 0$, y $\sum_{i=1}^{n} t_i = 1$. La densidad de esta distribución sigue la siguiente expresión:

$$\pi(\theta) \propto \prod_{i=1}^{n} \theta_i^{st_i - 1}$$

Esto determina la densidad hasta una constante de normalización. El valor esperado de esta densidad es igual a $t$. Este parámetro representa la creencia *a priori* acerca de los valores de probabilidad de la distribución multinomial. Si tenemos un conjunto de casos $D$ de $m$ casos independientes en los que observamos las siguientes frecuencias absolutas: $c = (c_1, \ldots, c_n)$ para los diferentes valores $(y_1, \ldots, y_n)$ de $Y$, entonces la distribución *a posteriori* es también una distribución de Dirichlet de los parámetros $(s + m, (s \cdot t + c)/(s + m))$.

Con esta distribución, el valor esperado de $\theta_i$ dado $D$ es

$$p_i^d = \frac{(s \cdot t_i + c_i)}{(s + m)}$$

y éste es el valor usado para estimar la probabilidad de $(Y = y_i)$ dado $D$.

Por consideraciones de simetría, en nuestro caso, los parámetros *a priori* $t$ son la distribución uniforme $t = (1/n, \ldots, 1/n)$. De esta forma la estimación de Dirichlet de las probabilidades para el conjunto $D$ es

$$p_i^d = \frac{(s/n + c_i)}{(s + m)}$$

.

El parámetro $s$ es el equivalente al tamaño de la muestra de los parámetros *a priori*[1]. Mayores valores de $s$ dan más importancia a las probabilidades *a priori* y menos a los datos de la muestra. Nótese que para $s = 0$ se le daría la máxima importancia a los datos de la muestra, es decir, la probabilidad se calcularía por máxima verosimilitud. En el caso de las probabilidades imprecisas, Walley [347] propone usar $s = 1$, aunque considera usar, en algunas situaciones, el valor $s = 2$. En nuestro caso, con una distribución *a priori* uniforme, al situación es distinta. Hemos encontrado que, en general, necesitamos mayores valores del parámetro $s$ para obtener los mejores resultados. Por tanto, hemos escogido $s = 2$. Para un valor fijo del parámetro y las mismas frecuencias relativas de los valores de $Y$, las probabilidades estimadas dependerán del tamaño de la muestra: muestras de tamaño más pequeño tenderán a producir distribuciones *más uniformes*, por ejemplo, distribuciones que tienen mayor entropía. Esto hace, por ejemplo, que cuando se ramifique por una variable de más valores se obtienen muestras más pequeñas en cada una de las hojas. Por tanto, las distribuciones en las hojas tiendan a ser más uniformes con menos ganancia de información y éste es un factor que perjudica la ramificación por variables de muchos casos.

Usando las probabilidades calculadas mediante la distribución de Dirichlet, se realiza el cálculo de la entropía $H_D^d(C)$ de $C$ dado $D$:

$$H_D^d(C) = \sum_{i=1}^{n} p_i^d log_2(1/p_i^d)$$

y, para cada variable, el cálculo de la entropía del nivel siguiente $H_D^d(X, D)$

$$H_D^d(C|X) = \sum_{j=1}^{k} p_D^d(X = x_j) H_{D_j}^d(C)$$

Una vez conocida la entropía $H_D^d(C)$ para la variable $C$ dado $D$ y, para cada variable $X$, la entropía que resultaría al ramificar por esa variable $H_D^d(C|X)$, escogemos aquella variable con menor entropía, es decir, la que introduce una menor incertidumbre en la expansión del árbol. O lo que es lo mismo, la que mayor ganancia de información produce.

[1] La idea intuitiva es que el tamaño de la muestra representa nuestra experiencia pasada con que los valores de $t$ fueron asignados.

El tratamiento de los valores perdidos que damos en este método es, básicamente, ignorarlos. Cuando el valor de una variable o clase es desconocido, simplemente no se incluye en el cómputo de las probabilidades.

La nueva condición de parada que se añade se basa en un principio muy simple: cuando se produzca un aumento de la entropía dejamos de ramificar, o lo que es lo mismo, paramos cuando obtenemos ganancias negativas, es decir, cuando todas las variables por las que podemos ramificar aumentan la incertidumbre ($H_D^d(C|X) \geq H_D^d(C)$). Téngase en cuenta que esto puede suceder aquí (pero no en probabilidades calculadas con máxima verosimilitud) debido a que en $H_D^d(C|X)$ usamos muestras más pequeñas y las distribuciones tienden a tener una mayor entropía.

No obstante, no hemos sido totalmente estrictos en la aplicación de esta condición de parada. Ya que, en algunos casos, se permite un nivel más: si tengo una ganancia negativa miro en el siguiente nivel, si tengo también ganancia negativa paro de ramificar. Es decir, si en un nivel tengo ganancia negativa y en el siguiente ganancia positiva, sigo ramificando.

La idea es admitir relaciones entre varias variables (relaciones que involucren a más de dos variables que no puedan ser detectadas cuando se observan parejas de variables), de forma que se acepta un nivel de aumento de la incertidumbre, para poder tratar las relaciones entre tres variables. De esta forma, si en el hijo con menor incertidumbre, la entropía es mayor a la del nodo padre y el nodo nieto disminuye la entropía, entonces se acepta el nodo hijo (y el nieto, en un nivel posterior).

## 4.3. Métodos de poda.

La construcción de los árboles de clasificación es, normalmente, un proceso de dos pasos. En un primer paso construimos el árbol y en un segundo proceso eliminamos (podamos) aquellas ramas que no nos interesen. La poda es un proceso de refinamiento que se introduce para evitar el sobreajuste al conjunto de datos, además de obtener, como resultado, árboles de menor tamaño y que generalicen mejor.

El tamaño de los árboles crece linealmente con el del conjunto de datos, aún cuando el rendimiento del árbol no aumenta de igual forma que su complejidad. Cuando entre los datos tenemos variables que son irrelevantes

(independientes de la variable a clasificar) también conviene eliminarlas del árbol.

La poda en los árboles de clasificación puede ser *ascendente* o *descendente*. En la poda descendente, empezamos en el nodo raíz y vamos cogiendo recursivamente distintas ramas con dirección a los nodos hoja. Cuando encontramos un subárbol que hay que eliminar, se poda y termina, en esa rama, el proceso de poda En la poda ascendente se empiezan en los nodos hojas y se va subiendo en la jerarquía del árbol, podándose aquellas ramas en caso de que sea necesario.

Existen distintos tipos de poda [107], entre los que destacamos los siguientes:

- *Reduced Error Pruning (REP)*: es un método de poda de los nodos hoja al nodo raíz (ascendente), propuesto por Quinlan en [278]. Es el método de poda más simple. Compara el error del conjunto de datos[2] de un nodo con el del subárbol que cuelga de él, si el error del padre es menor, se poda el subárbol hijo. Cada nodo sólo se visita una vez, por lo que es de orden lineal. Es el que hemos usado para el método basado en la distribución de Dirichlet y para ID3.

- *Pessimistic Error Pruning (PEP)*: propuesto por Quinlan en [278], introduce la corrección de continuidad para la distribución binomial en el calculo del error para obtener un error más realista. Es un método de poda descendente; compara el error del árbol con el guardado en el nodo, si el del nodo es menor, poda. No es un método muy ortodoxo, pero es muy rápido.

- *Minimum Error Pruning (MEP)*: es una estrategia descendente, que calcula la probabilidad esperada de que un caso alcance su clase correspondiente, en función de un parámetro $m$. El error se calcula en base a esta probabilidad [244]. Cuanto mayor es el parámetro $m$ mayor es la poda, por lo cual, la elección de dicho parámetro es muy importante y depende del problema.

[2]Entendemos por error de un conjunto de datos, al número de muestras de dicho conjunto de datos que se clasifican incorrectamente, dividido por el número total de muestras. Lo contrario a la precisión del clasificador.

- *Critical Value Pruning (CVP)*: es parecido a una técnica prepoda (en el sentido de que se puede aplicar mientras se construye el árbol). Se tiene un umbral, denominado *valor crítico* utilizado para decidir si un nodo se poda o no [234]. Por tanto, un nodo interno del árbol es podado si su medida del error es mayor al valor crítico. No obstante, si un nodo verifica este criterio pero no ocurre lo mismo con todos sus nodos hijos, no se realiza la poda pues contiene hijos relevantes. El grado de la poda varía con el valor crítico escogido. En [234] se realizan distintas podas con distintos valores de este umbral, para quedarse finalmente con el mejor árbol.

- *Cost-Complexity Pruning (CCP)*: también conocido como CART [53]. Se crea un conjunto de árboles, donde el árbol $T_{i+1}$ se obtiene a partir del árbol $T_i$ mediante la poda de algunos de sus nodos. La poda se hace en base a una medida que proponen, calculada en función del error y del número de nodos. Finalmente, se escoge el árbol con mayor precisión de todos los generados.

- *Error Based Pruning (EBP)*: propuesto por Quinlan en [279], es el método usado en el C4.5 y se considera una optimización del *Pessimistic Error Pruning*. Es ascendente. Para cada conjunto de ejemplos, se estima un intervalo de confianza[3], para modificar el error respecto a casos no vistos. Para cada nodo se tiene su error calculado de esta forma y el error que se obtendría si fuera podado, si este segundo error es menor, se realiza la poda.

Como ya hemos comentado, el método de construcción de árboles de clasificación utilizando una distribución de Dirichlet utiliza el método REP (*Reduced Error Pruning*), que es bastante sencillo y obtiene, como se verá, buenos resultados; también lo hemos aplicado a los árboles generados por ID3. C4.5 utiliza, en cambio, el esquema de EBP (*Error Based Pruning*).

| Base de Datos | Casos | Var. | Clases |
|---|---|---|---|
| Australian | 690 | 14 | 2 |
| Breast | 682 | 9 | 2 |
| Chess | 3196 | 36 | 2 |
| Cleve | 296 | 13 | 2 |
| Corral | 128 | 6 | 2 |
| Crx | 653 | 15 | 2 |
| Flare | 1066 | 10 | 2 |
| German | 1000 | 20 | 2 |
| Glass2 | 163 | 9 | 2 |
| Heart | 270 | 14 | 2 |
| Hepatitis | 80 | 19 | 2 |
| Iris | 150 | 4 | 3 |
| Lymphography | 148 | 18 | 4 |
| mofn_3_7_10 | 1324 | 10 | 2 |
| Pima | 768 | 8 | 2 |
| Segment | 2310 | 19 | 7 |
| shuttle_small | 5800 | 9 | 6 |
| soybean-large | 562 | 35 | 19 |
| Vote | 435 | 16 | 2 |
| Waveform_21 | 5000 | 21 | 3 |
| Monks1 | 556 | 6 | 2 |
| Monks2 | 601 | 6 | 2 |

Tabla 4.1: Descripción de las bases de datos utilizadas en los experimentos.

## 4.4. Resultados experimentales.

Dentro de este apartado haremos un estudio experimental de distintos métodos de construcción de árboles de clasificación descritos anteriormente: ID3, C4.5 y los árboles de clasificación construidos usando una estimación bayesiana (en adelante, llamaremos a este último tipo *Dirichlet*).

[3]En este caso el error se calcula $Error = m \cdot B_{fc}(e, n)$. Donde, $m$ es el número de casos, $e$ es el número de casos mal clasificados, $B$ es la función de distribución binomial y $fc$ es el factor de confianza. En C4.5 se suele tomar $fc = 0{,}25$.

Hemos aplicado dichos métodos sobre algunas bases de datos conocidas, obtenidas de *UCI repository of machine learning databases* [40]. En la tabla 4.1 hay una breve descripción de las bases de datos usadas: el campo *Casos* es el número de casos de cada problema; *Var.* es el número de variables; *Clases* es el número de valores diferentes que toma la variable a clasificar. Estas bases de datos han sido preprocesadas, aquellas que tenían variables continuas, fueron discretizadas usando el software MLC++[4]. La medida usada para discretizar fue la entropía. El número de intervalos no es fijo, y se obtiene siguiendo el procedimiento de Fayyad e Irani [111]. Los casos con valores perdidos fueron omitidos.

Por su ámbito de aplicación, las bases de datos provienen de la medicina: *Breast*, *Breast-cancer*, *Heart*, *Hepatitis*, *Cleveland*, *cleveland nominal* y *Pima*; del domino astrológico, *Flare1*; del campo de la política *Vote-irvine*; del campo financiero *German* y *Australian*; del campo botánico *Soybean-small* y *Soybean-large*; finalmente *Monks1* y *Monks2* son del dominio de la lógica (son bases de datos sintéticas).

Para los experimentos se ha usado *validación cruzada de 20-hojas* para calcular la bondad de cada método, esto es, el porcentaje de casos bien clasificados para cada uno de los problemas estudiados. También se ha utilizado el test de Wilcoxon de signos pareados por rangos para determinar si los resultados de los métodos son estadísticamente significativos.

En la tabla 4.2 podemos observar el tanto por ciento de bien clasificados para los métodos ID3 y Dirichlet, antes y después de la poda. También se ha mostrado el nivel de significación (que puede ser de 0,05 , 0,02 ó 0,01) para el caso de que sean estadísticamente diferentes o nada si no se puede demostrar que sean estadísticamente diferentes, según el test de Wilcoxon.

Como podemos ver, en media, el método basado en la estimación bayesiana obtiene mejores resultados, pero la diferencia no es excesivamente grande.

Observamos que Dirichlet es mejor en **26** casos (13 antes de podar y otros 13 después de la posa, mientras que ID3 es mejor en **15** casos (6 antes de la poda y 9 después de podar). Si nos fijamos en aquellas diferencias significa-

[4]Disponible en http://www.sgi.com/tech/mlc

| Problema | ID3 | Dirichlet | Sig. | ID3+REP | Dirichlet+REP | Sig. |
|---|---|---|---|---|---|---|
| Australian | $78,40 \pm 7,73$ | $81,60 \pm 7,45$ | 0,01 | $83,35 \pm 7,67$ | $85,37 \pm 6,58$ | No |
| Breast | $91,08 \pm 3,12$ | $91,95 \pm 3,18$ | No | $91,08 \pm 3,13$ | $94,23 \pm 3,53$ | 0,01 |
| Chess | $99,72 \pm 0,51$ | $99,66 \pm 0,58$ | No | $97,87 \pm 2,29$ | $98,56 \pm 1,40$ | No |
| Cleve | $78,64 \pm 8,72$ | $80,38 \pm 8,71$ | No | $79,67 \pm 7,20$ | $78,00 \pm 8,29$ | No |
| Corral | $100,00 \pm 0,00$ | $100,00 \pm 0,00$ | No | $92,02 \pm 13,99$ | $84,29 \pm 13,18$ | 0,01 |
| Crx | $79,62 \pm 7,39$ | $83,75 \pm 6,71$ | 0,02 | $85,77 \pm 6,84$ | $86,52 \pm 6,01$ | No |
| Flare | $82,27 \pm 3,08$ | $82,08 \pm 3,26$ | No | $82,27 \pm 3,08$ | $82,74 \pm 1,84$ | No |
| German | $66,40 \pm 4,67$ | $68,80 \pm 5,88$ | No | $70,30 \pm 2,78$ | $70,20 \pm 0,87$ | No |
| Glass2 | $84,79 \pm 11,34$ | $84,79 \pm 12,01$ | No | $84,17 \pm 12,18$ | $84,79 \pm 12,01$ | No |
| Heart | $82,97 \pm 6,57$ | $83,68 \pm 7,69$ | No | $79,62 \pm 7,44$ | $79,67 \pm 7,60$ | No |
| Hepatitis | $87,50 \pm 16,77$ | $90,00 \pm 14,58$ | No | $87,50 \pm 16,77$ | $90,00 \pm 14,58$ | No |
| Iris | $93,13 \pm 8,25$ | $95,89 \pm 6,29$ | No | $93,13 \pm 8,25$ | $95,18 \pm 7,99$ | No |
| Lymphography | $76,61 \pm 16,69$ | $74,91 \pm 14,72$ | No | $75,80 \pm 16,68$ | $74,73 \pm 16,06$ | No |
| Mofn_3_7_10 | $100,00 \pm 0,00$ | $100,00 \pm 0,00$ | No | $83,15 \pm 5,66$ | $82,33 \pm 4,21$ | No |
| Pima | $76,17 \pm 5,93$ | $76,83 \pm 6,12$ | No | $76,18 \pm 6,48$ | $77,75 \pm 7,69$ | No |
| Segment | $93,98 \pm 2,26$ | $92,77 \pm 2,40$ | 0,05 | $94,07 \pm 2,32$ | $92,68 \pm 2,45$ | 0,02 |
| Shuttle_small | $99,76 \pm 0,27$ | $99,66 \pm 0,33$ | No | $99,76 \pm 0,27$ | $99,66 \pm 0,33$ | No |
| Soybean_large | $89,31 \pm 5,55$ | $91,65 \pm 4,37$ | No | $89,13 \pm 5,60$ | $91,47 \pm 4,54$ | No |
| Vote | $93,36 \pm 5,83$ | $94,51 \pm 5,69$ | No | $93,57 \pm 4,87$ | $94,29 \pm 5,56$ | No |
| Waveform_21 | $70,36 \pm 2,85$ | $75,34 \pm 2,82$ | 0,01 | $71,80 \pm 2,93$ | $74,28 \pm 2,76$ | 0,01 |
| Monks1 | $98,74 \pm 3,25$ | $86,52 \pm 6,91$ | 0,01 | $86,71 \pm 7,61$ | $82,89 \pm 6,48$ | 0,02 |
| Monks2 | $69,59 \pm 8,94$ | $73,40 \pm 8,40$ | 0,01 | $66,09 \pm 5,84$ | $65,73 \pm 1,46$ | No |
| **Media** | $\mathbf{86,02 \pm 5,89}$ | $\mathbf{86,75 \pm 5,82}$ | | $\mathbf{84,68 \pm 6,81}$ | $\mathbf{84,79 \pm 6,16}$ | |

Tabla 4.2: Precisión en tanto por ciento de ID3 y Dirichlet usando validación cruzada de 20-hojas.

tivas estadísticamente, Dirichlet es estadísticamente diferente y mejor a ID3 en **6** casos (4 antes de podar y 2 después de la poda), mientras que ID3 es mejor en **5** casos (2 antes de la poda y 3 después de la poda).

En la tabla 4.3 podemos observar el tanto por ciento de bien clasificados para los métodos C4.5 y Dirichlet, antes y después de la poda. Como podemos ver, en media, el método Dirichlet obtiene mejores resultados que C4.5, al igual que pasaba con ID3.

Resumiendo, Dirichlet es mejor a C4.5 en **28** casos (13 antes de podar y 15 después de la poda), mientras que C4.5 es sólo mejor en **12** ejecuciones (6 antes de la poda, 6 después). Si nos fijamos exclusivamente los resultados significativos estadísticamente, Dirichlet es mejor a C4.5 en **7** casos (2 antes de podar y 5 después de la poda), mientras que C4.5 es mejor sólo en **5** casos (2 antes de la poda y 3 después de la poda).

En las tablas 4.4 y 4.5 mostramos el número de nodos de cada árbol ge-

| Problema | C4.5 | Dirichlet | Sig. | C4.5+EBP | Dirichlet+REP | Sig. |
|---|---|---|---|---|---|---|
| Australian | $79,97 \pm 7,30$ | $81,60 \pm 7,45$ | No | $85,51 \pm 6,90$ | $85,37 \pm 6,58$ | No |
| Breast | $90,48 \pm 3,33$ | $91,95 \pm 3,18$ | No | $91,35 \pm 4,11$ | $94,23 \pm 3,53$ | 0,05 |
| Chess | $99,69 \pm 0,51$ | $99,66 \pm 0,58$ | No | $94,34 \pm 1,37$ | $98,56 \pm 1,40$ | 0,01 |
| Cleve | $77,29 \pm 13,92$ | $80,38 \pm 8,71$ | No | $75,31 \pm 9,92$ | $78,00 \pm 8,29$ | No |
| Corral | $100,00 \pm 0,00$ | $100,00 \pm 0,00$ | No | $75,60 \pm 15,93$ | $84,29 \pm 13,18$ | No |
| Crx | $78,99 \pm 7,84$ | $83,75 \pm 6,71$ | 0,01 | $86,39 \pm 6,48$ | $86,52 \pm 6,01$ | No |
| Flare | $81,89 \pm 2,84$ | $82,08 \pm 3,26$ | No | $83,21 \pm 1,94$ | $82,74 \pm 1,84$ | No |
| German | $66,30 \pm 5,52$ | $68,80 \pm 5,88$ | No | $70,00 \pm 0,00$ | $70,20 \pm 0,87$ | No |
| Glass2 | $84,17 \pm 12,80$ | $84,79 \pm 12,01$ | No | $62,08 \pm 13,55$ | $84,79 \pm 12,01$ | 0,01 |
| Heart | $83,35 \pm 7,58$ | $83,68 \pm 7,69$ | No | $74,84 \pm 7,92$ | $79,67 \pm 7,60$ | No |
| Hepatitis | $87,50 \pm 16,77$ | $90,00 \pm 14,58$ | No | $87,50 \pm 14,79$ | $90,00 \pm 14,58$ | No |
| Iris | $93,13 \pm 8,25$ | $95,89 \pm 6,29$ | No | $94,01 \pm 6,64$ | $95,18 \pm 7,99$ | No |
| Lymphography | $76,43 \pm 15,73$ | $74,91 \pm 14,72$ | No | $81,61 \pm 14,90$ | $74,73 \pm 16,06$ | 0,01 |
| Mofn_3_7_10 | $100,00 \pm 0,00$ | $100,00 \pm 0,00$ | No | $77,95 \pm 0,91$ | $82,33 \pm 4,21$ | 0,01 |
| Pima | $76,17 \pm 5,77$ | $76,83 \pm 6,12$ | No | $76,84 \pm 7,20$ | $77,75 \pm 7,69$ | No |
| Segment | $93,98 \pm 2,26$ | $92,77 \pm 2,40$ | No | $94,33 \pm 1,48$ | $92,68 \pm 2,45$ | 0,01 |
| Shuttle_small | $99,79 \pm 0,32$ | $99,66 \pm 0,33$ | 0,01 | $99,55 \pm 0,35$ | $99,66 \pm 0,33$ | No |
| Soybean_large | $91,11 \pm 4,85$ | $91,65 \pm 4,37$ | No | $92,18 \pm 4,29$ | $91,47 \pm 4,54$ | 0,05 |
| Vote | $92,68 \pm 5,43$ | $94,51 \pm 5,69$ | No | $94,06 \pm 5,41$ | $94,29 \pm 5,56$ | No |
| Waveform_21 | $70,00 \pm 2,88$ | $75,34 \pm 2,82$ | 0,01 | $74,82 \pm 2,20$ | $74,28 \pm 2,76$ | No |
| Monks1 | $97,48 \pm 4,71$ | $86,52 \pm 6,91$ | 0,01 | $74,78 \pm 5,01$ | $82,89 \pm 6,48$ | 0,01 |
| Monks2 | $73,40 \pm 8,43$ | $73,40 \pm 8,40$ | No | $65,73 \pm 1,46$ | $65,73 \pm 1,46$ | No |
| **Media** | $\mathbf{86,08 \pm 6,23}$ | $\mathbf{86,75 \pm 5,82}$ | | $\mathbf{82,36 \pm 6,04}$ | $\mathbf{84,79 \pm 6,16}$ | |

Tabla 4.3: Precisión en tanto por ciento de los métodos C4.5 y Dirichlet usando validación cruzada de 20-hojas.

nerados por los distintos métodos, antes de podar y después de podar. Si observamos la tabla 4.4, antes de la poda, vemos que el número de nodos generado por el método de Dirichlet es bastante menor que los generados por ID3 o C4.5 (menos de la mitad). El tanto por ciento de bien clasificados antes de la poda es también mejor a ID3 y a C4.5. Podemos decir que genera árboles que generalizan mejor el conjunto de datos. Además, al generar un número menor de nodos, la construcción del árbol es más rápida (como vemos en la tabla 4.7). En la tabla 4.5 observamos que el número de nodos, después de la poda, en Dirichlet es un poco mayor que C4.5, pero una diferencia apenas apreciable y bastante menor que ID3, que utiliza igual método de poda.

El hecho de utilizar distinto método de poda para Dirichlet e ID3 frente a C4.5, puede dar lugar a pensar que la comparativa ha sido desigual. En la tabla 4.6, sólo con valores medios, podemos observar como con el método REP el algoritmo C4.5 obtiene peores resultados tras la poda. También,

| Problema | ID3 | C4.5 | Dirich. |
|---|---|---|---|
| Australian | 595 | 559 | 195 |
| Breast | 211 | 218 | 218 |
| Chess | 95 | 99 | 79 |
| Cleve | 218 | 224 | 137 |
| Corral | 27 | 27 | 27 |
| Crx | 499 | 544 | 206 |
| Flare | 719 | 617 | 215 |
| German | 963 | 998 | 638 |
| Glass2 | 81 | 70 | 45 |
| Heart | 217 | 213 | 117 |
| Hepatitis | 27 | 23 | 19 |
| Iris | 31 | 31 | 7 |
| Lymphography | 94 | 101 | 33 |
| Mofn_3_7_10 | 111 | 111 | 111 |
| Pima | 322 | 317 | 150 |
| Segment | 1157 | 1162 | 196 |
| Shuttle_small | 140 | 175 | 50 |
| Soybean-large | 231 | 207 | 76 |
| Vote | 73 | 82 | 28 |
| Waveform_21 | 3724 | 3736 | 686 |
| Monks1 | 192 | 124 | 33 |
| Monks2 | 454 | 443 | 446 |
| **Media** | **462** | **458** | **168** |

Tabla 4.4: Número de nodos de los árboles de clasificación, sin podar, para los distintos métodos

podemos ver, como Dirichlet e ID3 obtienen peores resultados al ser podados con EBP.

En la tabla 4.7 observamos la suma de los tiempos, en milisegundos, que han tardado en construir cada método estudiado el árbol de clasificación, teniendo en cuenta todos los problemas presentados (ver tabla 6.2). Observando la tabla 4.7, de tiempos, confirmamos lo que se había dicho con anterioridad del método Dirichlet, al generar menos nodos y ser más simple en su concepción, tarda sobre la mitad de tiempo que los algoritmos ID3 y C4.5, tanto si se aplica el proceso de la poda como si no se aplica.

| Problema | ID3+REP | C4.5+EBP | Dirich.+REP |
|---|---|---|---|
| Australian | 82 | 3 | 12 |
| Breast | 211 | 80 | 31 |
| Chess | 67 | 13 | 47 |
| Cleve | 72 | 37 | 84 |
| Corral | 27 | 3 | 3 |
| Crx | 3 | 3 | 8 |
| Flare | 660 | 3 | 102 |
| German | 1 | 1 | 1 |
| Glass2 | 81 | 3 | 36 |
| Heart | 79 | 17 | 45 |
| Hepatitis | 27 | 1 | 11 |
| Iris | 13 | 13 | 4 |
| Lymphography | 94 | 26 | 29 |
| Mofn_3_7_10 | 23 | 1 | 67 |
| Pima | 300 | 34 | 20 |
| Segment | 1157 | 330 | 196 |
| Shuttle_small | 140 | 175 | 50 |
| Soybean-large | 231 | 77 | 76 |
| Vote | 73 | 4 | 22 |
| Waveform_21 | 2517 | 493 | 477 |
| Monks1 | 113 | 5 | 17 |
| Monks2 | 324 | 1 | 1 |
| **Media** | **286** | **54** | **60** |

Tabla 4.5: Número de nodos de los árboles de clasificación podados, para los distintos métodos

| Problema | ID3 | C4.5 | Dirich. |
|---|---|---|---|
| Sin poda | 86,02 % | 86,08 % | 86,75 % |
| Usando REP | 84,68 % | 81,97 % | 84,79 % |
| Usando EBP | 82,96 % | 82,36 % | 82,98 % |

Tabla 4.6: Precisión media de los árboles de clasificación, para distintos métodos de poda.

| Problema | Tiempos |
|---|---|
| ID3 | 2396 milisegundos |
| C4.5 | 2430 milisegundos |
| Dirichlet | 1284 milisegundos |
| ID3+REP | 2546 milisegundos |
| C4.5+EBP | 2666 milisegundos |
| Dirichlet+REP | 1382 milisegundos |

Tabla 4.7: Tiempos en milisegundos empleado por cada algoritmo en todos los problemas tratados, tanto si no se aplican los método con poda, como en el caso de que se apliquen.

## 4.5. Discusión.

En este capítulo se ha presentado un nuevo método para la construcción de árboles de clasificación y se ha comparado con otros métodos clásicos, como son ID3 y C4.5. El método desarrollado es una simplificación de estos algoritmos clásicos, introduciendo una variante: usamos una distribución *a priori* de Dirichlet para el cálculo de la probabilidad que se usa en las entropías. Hemos visto que con esta modificación y simplificación, se han mejorado los resultados en precisión y en tiempo. Podemos resumir las conclusiones, en los siguientes puntos:

- El método que hemos presentado es más simple pues sólo utiliza el concepto de ganancia, en lugar de la razón de ganancia (C4.5). Además se añade una nueva condición de parada, que genera árboles más pequeños.
- Se han estudiado distintos métodos de poda existentes, escogiendo aquellos que mejor se adaptaban a los resultados de cada método. En el método que presentamos se ha escogido *Reduced Error Pruning* al igual que con ID3, que es un algoritmo de poda más simple que el escogido para C4.5, que es *Error Based Pruning*.
- La precisión en la clasificación, en media, es peor en los métodos clásicos que en el método propuesto.
- El número de nodos de un árbol de clasificación construido usando

una distribución de Dirichlet es bastante menor en la fase de construcción. Al generar menos nodos, podemos deducir que es más rápido que los otros métodos clásicos en la construcción del árbol. Esto es debido a la nueva condición de parada utilizada. Así, además de un mejor rendimiento por disminución de complejidad, obtenemos un mejor rendimiento al ramificar menos.

# Capítulo 5

# Multiredes bayesianas como clasificadores.

Las multiredes bayesianas son una extensión de las redes bayesianas donde es posible representar independencias condicionales dependientes del contexto.

En las multiredes tenemos una variable distinguida y una red bayesiana para cada valor que pueda tomar dicha variable. De forma intuitiva podemos ver una multired bayesiana como un árbol de clasificación de profundidad uno, con redes bayesianas en las hojas. Al igual que los clasificadores bayesianos o los árboles de clasificación, podemos utilizar las multiredes bayesianas para clasificación supervisada.

Cuando hablamos de multiredes bayesianas para el problema de la clasificación supervisada, principalmente, podemos distinguir dos tipos: aquellas en las que la variable distinguida es la clase y un segundo tipo donde la variable distinguida es un atributo.

Dentro de este segundo tipo podemos seguir eligiendo variables de forma recursiva obteniendo una estructura de árbol en donde las hojas son redes bayesianas. Este segundo tipo se denomina multired bayesiana recursiva.

## 5.1. Independencias asimétricas.

Las redes bayesianas nos permiten representar independencias entre variables. En teoría de la probabilidad, la independencia de $X$ e $Y$ dado $Z$

implica que $P(X, Y|Z) = P(X|Z)$ para todos los valores de $X$,$Y$ y $Z$. En particular, la independencia se cumple para todos los valores de $Z$ (simétrica en todos los casos). Basta con que para un caso $Z = z$ esa igualdad no se soporte para que $X$ e $Y$ no sean considerados independientes dado $Z$, aunque lo sean para el resto de casos.

Las dependencias e independencias codificadas por una red bayesiana son *independencias condicionales independientes del contexto* (en inglés, *context-non-specific conditional (in)dependencies*) [335]. Por otro lado, las independencias donde se soporta sólo para algunas instancias de sus variables, se conocen como *independencias condicionales dependientes del contexto* (del inglés, *context-specific conditional independencies*)[128], también denominadas *independencias condicionales asimétricas* o *independencias de contexto específico* [51].

Para ver claramente el significado de este tipo de independencias veamos algún ejemplo. Supongamos que estamos estudiando el cáncer de mama en base a una serie de variables, es evidente que dependiendo del sexo de los pacientes nos saldrán dos distribuciones de probabilidad bastante distintas. Otro ejemplo, sería la variable fumador en el análisis del cáncer de pulmón: la mayoría de pacientes que padece este tipo de enfermedad son fumadores o exfumadores.

Hay distintos formalismos para poder capturar este tipo de independencias, que aumentan el poder de representación de las redes bayesianas, a la vez que mejoraran su eficiencia en la inferencia. Algunos de estos formalismos son las *redes de similaridad* [147] o las *multiredes bayesianas* [128].

El formalismo de las redes de similaridad propuesto en [147] está compuesto por un *grafo de similaridad*, y una colección de redes bayesianas, que el autor denomina *mapas de conocimiento local*. En el grafo de similaridad, cada nodo representa una o varias clases, mientras que los enlaces conectan clases que el experto considera similares o difícilmente distinguibles. Por cada enlace del grafo de similaridad, se tiene una red bayesiana. En cada mapa de conocimiento local se incluyen aquellas variables que ayuden a discriminar entre las clases pertenecientes al correspondiente enlace del grafo de similaridad. De esta forma se descompone el problema para trabajar con el experto.

En este formalismo las independencias asimétricas vienen representadas en los diversos mapas de conocimiento local, ya que para diferentes valores

de la clase, podemos tener distintas redes bayesianas.

Por ejemplo, para un problema con cuatro variables $\{X, Y, Z, C\}$, se muestra una posible representación de una red de similaridad en la figura 5.1, donde la variable clase $C$ tiene tres casos, $c_1, c_2, c_3$.

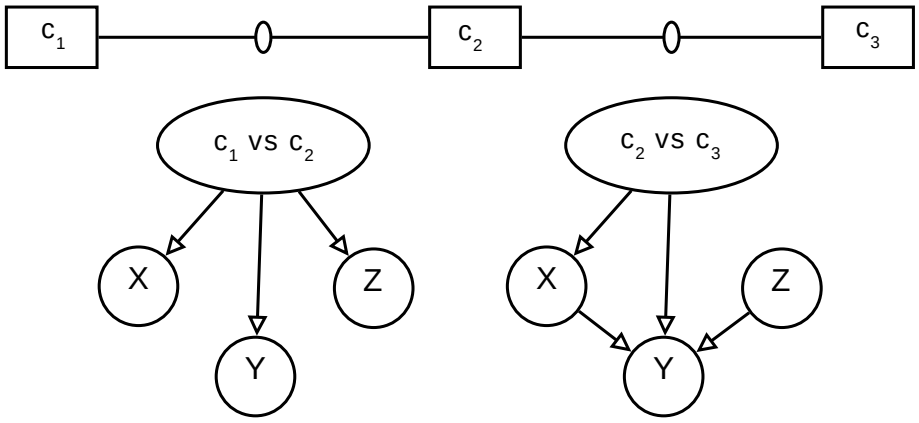


Figura 5.1: Red de similaridad: grafo de similaridad + redes bayesianas

Las multiredes bayesianas [128] también permiten representar este tipo de independencias. Para ello, tienen una variable distinguida que particiona el espacio en función de sus valores, obteniendo una red bayesiana para cada uno de estos valores, como puede verse en la figura 5.2.

Igual a como sucede con las redes de similaridad, las redes locales están restringidas a un subconjunto específico de casos y las relaciones de independencia asimétrica se representan en la topología de las redes locales. Estas independencias se manifiestan cuando un enlace entre atributos está presente en alguna red local pero no en otras. Otras independencias asimétricas se representan cuando existe un enlace clase-atributo en alguna red local pero no en otras. Con las multiredes es posible realizar cualquier tipo de inferencia.

## 5.2. Multiredes bayesianas.

Una multired bayesiana (abreviadamente BMN del inglés, *Bayesian Multinet*) es un conjunto de redes bayesianas, una para cada posible valor de

la variable distinguida. Más formalmente, para un conjunto de variables $\mathbf{Y} = (\mathbf{X}, Z)$ donde $\mathbf{X} = (X_1, ..., X_n)$ y $Z$ es la variable distinguida. Una multired bayesiana es una factorización gráfica de la distribución de probabilidad de $\mathbf{Y}$. La multired bayesiana se define a partir de la distribución de probabilidad de la variable $Z$ y un conjunto de redes bayesianas componentes para $\mathbf{Y} \setminus \{Z\}$, cada una de las cuales codifica la distribución de probabilidad conjunta para $\mathbf{Y} \setminus \{Z\}$ dado un valor de la variable $Z$. De esta forma sería posible obtener la distribución de probabilidad para $\mathbf{Y}$ representada por la multired de la siguiente forma: sea $P(Z)$ la distribución de probabilidad de $Z$ en la multired, entonces la distribución conjunta de $Y$ representada por la multired, es de la forma:

$$P(X_1, \ldots, X_n, z_i) = P_{B_i}(X_1, \ldots, X_n) \cdot P(Z = z_i)$$

donde $B_i$ es la red bayesiana para $Z = z_i$.

Heckerman en [147] distingue entre dos tipos de independencias asimétricas, las que denomina *de subconjunto* (cuando se trata de una relación entre la variable a clasificar y los atributos) y las de *de hipótesis específica* (cuando se trata de una relación entre atributos únicamente).

Por tanto, dentro de las multiredes para clasificación podemos distinguir dos tipos: aquellas en las que la variable distinguida es la variable a clasificar, donde se construirá una red bayesiana para cada clase, como muestra la figura 5.2. Este tipo de multired pretende modelizar las *independencias asimétricas de subconjunto*. En el segundo tipo tenemos las que la variable distinguida es un atributo, como podemos ver en la figura 5.3; este segundo subtipo pretende modelizar las *independencias asimétricas de hipótesis específica*. En este trabajo nos vamos a centrar en las multiredes cuya variable distinguida es un atributo. A este tipo de multiredes también se les conoce por *mixturas de redes bayesianas* [335].

### 5.2.1. Multiredes bayesianas recursivas.

Las *Multiredes Bayesianas Recursivas (rBMN)* [270] extienden las redes bayesianas y las multiredes bayesianas, ya que nos permiten más de un nivel de decisión en el árbol, es decir, podemos definir las rBMN haciendo uso de

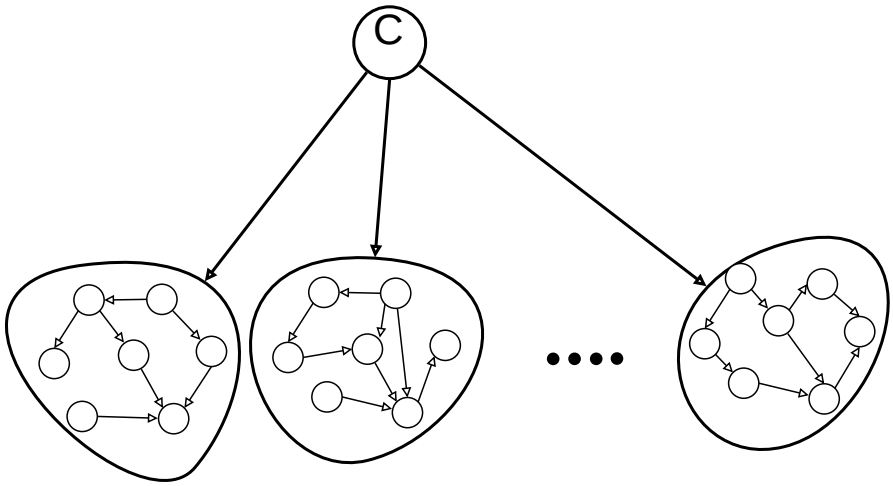


Figura 5.2: Multired bayesiana con la clase como variable distinguida

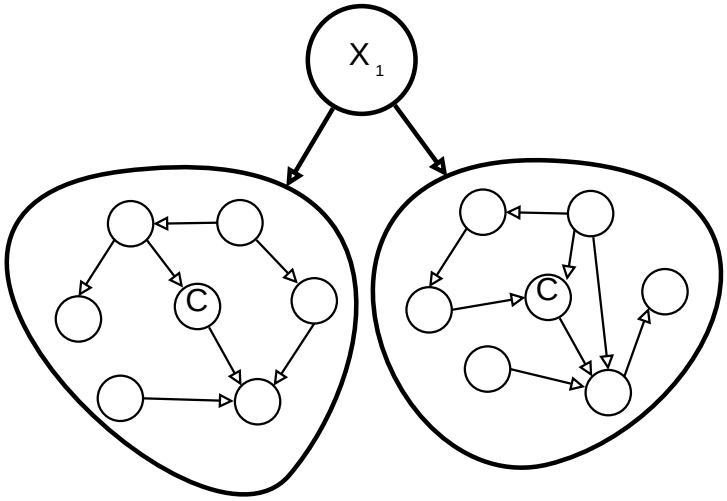


Figura 5.3: Multired bayesiana con un atributo como variable distinguida

la idea intuitiva de un árbol de clasificación con redes bayesianas en las hojas.

Sea **T** un árbol de clasificación, que llamaremos *árbol de clasificación distinguido*, que verifica:

1. Cada nodo interno de $\mathbf{T}$ representa una variable de $\mathbf{Y}$.

2. Cada nodo interno tendrá tantos hijos o ramas como estados tenga la variable que representa.

3. Si $\mathbf{T}(raiz, l)$ es el conjunto de variables que están en el camino de decisión entre la $raiz$ y el nodo hoja $l$ del árbol de clasificación distinguido, entonces no hay variables repetidas en $\mathbf{T}(raiz, l)$.

Un ejemplo de rBMN puede verse en la figura 5.4.

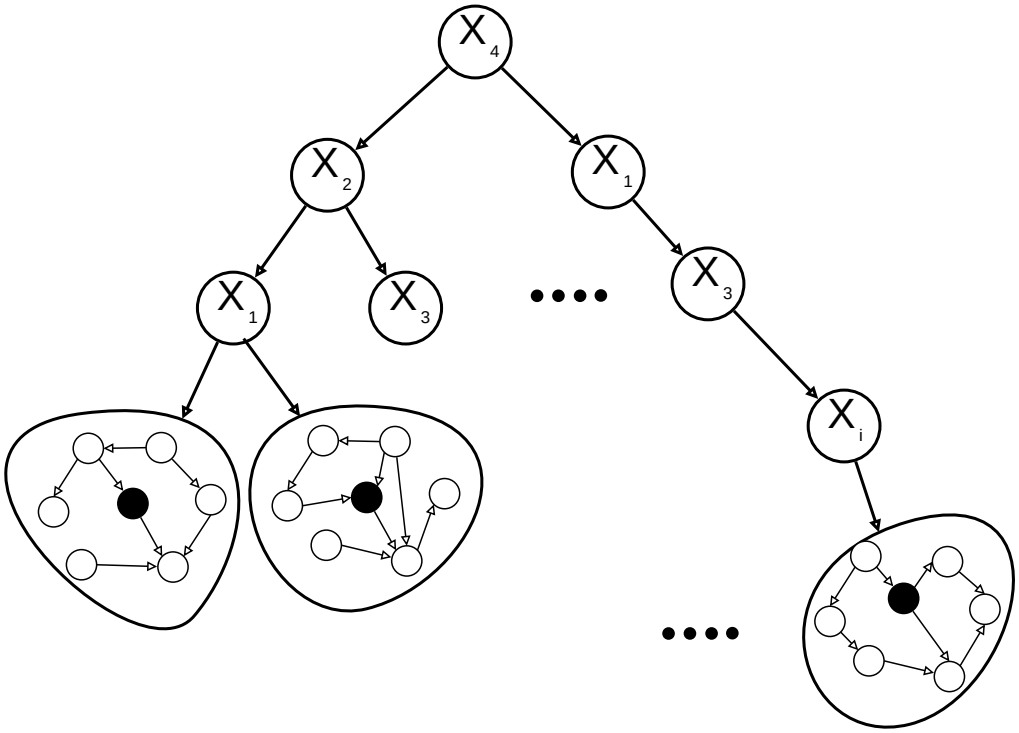


Figura 5.4: Multired bayesiana recursiva.

Definamos $\mathbf{Y} \setminus \mathbf{T}(raiz, l)$ como el conjunto de todas las variables en $\mathbf{Y}$ excepto aquellas que están en el camino de decisión entre la $raiz$ y el nodo hoja $l$ del árbol de clasificación distinguido. Entonces una rBMN para un conjunto de variables $\mathbf{Y} = (X_1, ..., X_n)$ y un árbol de clasificación distinguido $\mathbf{T}$, es una factorización gráfica de la distribución de probabilidad para $\mathbf{Y}$

que está definida por la distribución de probabilidad de las hojas de $\mathbf{T}$ y un conjunto de redes bayesianas. una por cada hoja de $\mathbf{T}$. Cada una de estas redes bayesianas codifica la distribución de probabilidad para $\mathbf{Y} \setminus \mathbf{T}(raiz, l)$ dada la hoja $l$ de $\mathbf{T}$. De esta forma las redes bayesianas en cada nodo hoja del árbol no contienen los atributos usados en el camino desde la raíz del árbol.

## 5.3. Multiredes bayesianas de filtrado y de envoltura.

En las multiredes bayesianas donde la variable distinguida es un atributo, el principal problema es encontrar cuál es la mejor variable por la cual se van a obtener las distintas redes bayesianas. La construcción de la multired va a conllevar un proceso de búsqueda para la variable distinguida, con el objetivo de maximizar la precisión del clasificador final y así, obtener un mejor modelo.

Para ello vamos a seguir dos enfoques ya comentados: envoltura y filtrado.

### 5.3.1. Multired bayesiana de envoltura ($BMN_w$).

Para encontrar la variable distinguida mediante este enfoque se van a seguir los siguientes pasos:

- Para cada atributo $X_i$ se calcula la precisión ponderada de los clasificadores que generaría si se expandiese por el nodo $X_i$. La precisión se calculará usando una validación cruzada de 5-hojas.
- Se escoge la variable $X_j$ con mayor bondad para hacer la ramificación.
- Divide el conjunto de casos $D$ de acuerdo a los valores de $X_j$. Construye un clasificador por cada valor de $X_j$.

Hay que tener en cuenta que este algoritmo construye todas las posibles multiredes bayesianas y evalúa cada una de ellas, por tanto, como se puede suponer es bastante costoso en tiempo. Por este motivo sólo vamos a estudiar un tipo de multired bayesiana de envoltura, la cual usará naïve bayes

en las hojas ($BMN_wNB$), ya que este clasificador es muy rápido de construir.

El clasificador *Naïve-Bayes Tree (NBTree)* [195], como vimos, combina las ideas de árbol de clasificación y naïve bayes. Teniendo naïve bayes en las hojas del árbol de clasificación, también se puede ver como una multired recursiva con naïve bayes en las hojas. La multired de envoltura que vamos a usar en nuestros experimentos es igual al NBTree pero sólo explora un nivel, es decir, no es recursiva.

Nótese que estamos usando clasificadores naïve bayes para medir la bondad de la multired construida y, por tanto, seleccionar que atributo es la mejor variable distinguida. Por tanto, no tendría sentido, una vez seleccionada la variable distinguida utilizar otro tipo de clasificador bayesiano en las hojas (como un clasificador TAN). Esto es así porque lo que en realidad estamos optimizando es la variable distinguida sólo cuando estamos utilizando naïve bayes en las hojas y no otro clasificador. No obstante, veremos que las funciones de filtrado son independientes del clasificador que utilicemos en las hojas y sí nos van a permitir utilizar el clasificador que queramos en las hojas.

### 5.3.2. Multired bayesiana de filtrado ($BMN_f$).

En las multiredes bayesianas de filtrado, utilizamos una función (que denominaremos *filtro*) que calcula para cada atributo una medida de su bondad para producir la ramificación. Algunas de las funciones que hemos escogido ya han sido tratadas en la literatura y otras son heurísticas que proponemos.

A partir del trabajo de Ben-Bassat [31] hemos usado tres medidas de filtrado: Kullback-Leibler, Matusita y Bhattacharyya. Estas métricas tenían la dificultad de que estaban diseñadas para problemas dicotómicos (donde la clase sólo tenía dos estados). No obstante, Armañanzas en [15] muestra estas medidas para problemas con más de dos clases.

**Filtro divergencia de Kullback-Leibler ($BMN_{f1}$)** La divergencia de Kullback-Leibler [31] es el método más conocido para la medición de distancias entre dos distribuciones de probabilidad. Su formulación genérica es:

$$D_{kl}\left(P(X), Q(X)\right) = \sum_{x_i} p(x_i) \log \frac{p(x_i)}{q(x_i)}$$

Para problemas con más de dos clases [15]:

$$KL(X,C) = \sum_{i=1}^{m} \sum_{j=1}^{j<i} P(c_i)P(c_j)\ KL_{ij}(X,C)_2$$

donde $m$ es el número de valores que toma la variable clase $C$ y $KL_{ij}(X,C)_2$ se define [15]:

$$KL_{ij}(X,C)_2 = D_{kl}\left(P(X|c_i), P(X|c_j)\right) + D_{kl}\left(P(X|c_j), P(X|c_i)\right)$$

**Filtro distancia de Matusita ($BMN_{f2}$)** La definición original de esta medida [31] mide la distancia entre dos distribuciones de probabilidad. Al generalizarse [15], se intenta medir la distancia media entre las diferentes distribuciones marginales de cada uno de los valores del atributo con los valores de la clase. Su expresión matemática se formula como:

$$D_m(X,C) = \sum_{i=1}^{m} \sum_{j=1}^{j<i} p(c_i)p(c_j) \left[ \sum_{t=1}^{k} \sqrt{p(x_t|c_i)p(x_t|c_j)} \right]$$

donde $k$ es el número de valores distintos que toma la variable $X$.

**Filtro ganancia ($BMN_{f3}$)** Esta función calcula la ganancia de información al igual que vimos en el algoritmo ID3 [279] con los árboles de clasificación, para una variable $X$ y conjunto de datos $D$.

$$ganancia(C|X) = H_D(C) - H_D(C|X)$$

donde la entropía es:

$$H_D(C) = \sum_{i=1}^{m} p_i log_2(1/p_i)$$

y la entropía media (generada por el atributo $X$) se calcula en el siguiente nivel:

$$H_D(C|X) = \sum_{j=1}^{k} p(X = x_j) H_{D_j}(C)$$

**Filtro razón de ganancia ($BMN_{f4}$)** Esta medida calcula la razón de ganancia como vimos que utilizaba el algoritmo C4.5 [279] en los árboles de clasificación. Además de la ganancia, calcula la información de ruptura, y usando las dos medidas anteriores, obtiene la razón de ganancia. Para un atributo $X$ la razón de ganancia se define como sigue:

$$RazónGanancia(X) = \frac{ganancia(C|X)}{InfoRuptura(X)}$$

donde,

$$InfoRuptura(X) = \sum_{j=1}^{k} p(X = x_j) \log_2(1/p(X = x_j))$$

**Filtro heurístico *Atributo padre del resto*** En esta heurística construimos una red bayesiana para cada atributo $X_i$, donde colocamos este atributo como padre de todas las variables, incluida la clase. Todas las variables, excepto $X_i$ forman una estructura naïve bayes, como podemos ver en la figura 5.5. Una vez construida la red calculamos su bondad con una métrica bayesiana; en nuestros experimentos hemos probado tres métricas K2 ($BMN_{f5}$), BIC ($BMN_{f6}$) y BDe ($BMN_{f7}$). El motivo detrás de esta heurística, es que esta estructura es equivalente a una multired de naïve bayes, como se puede adivinar en la estructura representada por la figura 5.5. Esta función de filtrado también puede adolecer del mismo defecto que le achacábamos a la versión de envoltura, esto es, sólo sería ideal su aplicación a multiredes bayesianas donde tenemos naïve bayes en las hojas.

**Filtro heurístico *padre o no de la clase*** En este caso se ha calculado para cada variable $X_i$ la métrica de una red bayesiana donde el atributo es padre de la clase (sin tener en cuenta el resto de atributos, como se puede ver en la figura 5.6(a)), menos el valor de la métrica de la variable clase sola (ver figura 5.6(b)). En esta medida también usamos las siguientes métricas bayesianas: K2 ($BMN_{f8}$), BIC ($BMN_{f9}$) y BDe ($BMN_{10}$).

**Filtro información mutua condicionada ($BMN_{f11}$)** En esta medida calculamos para cada variable $X_i$ y para el resto de variables $X_j$ tales que $i \neq j$, la información mutua condicionada dada la clase. De esta forma, para $X_i$ obtenemos la suma de la información mutua condicionada dada la clase con el resto de los atributos, quedándonos con aquella que más información

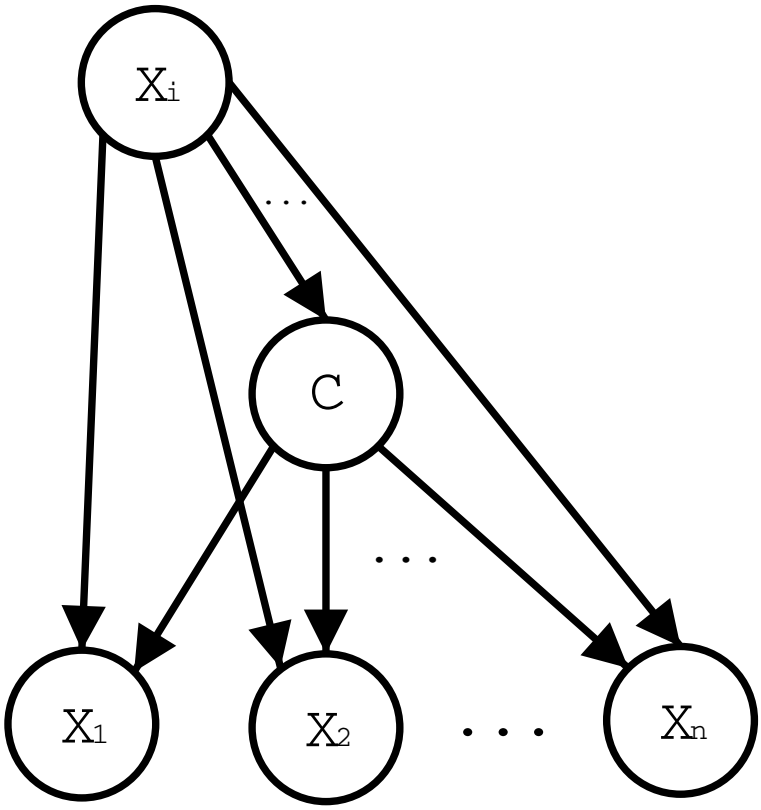


Figura 5.5: Estructura usada en la *heurística atributo padre del resto.*

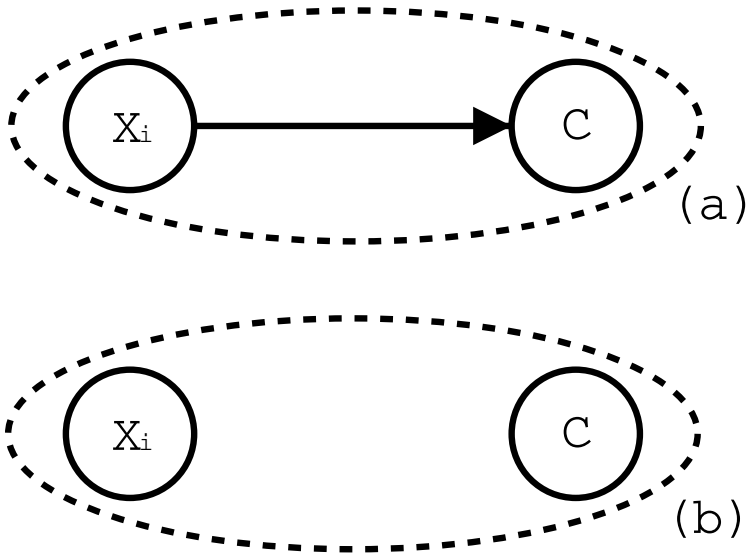


Figura 5.6: (a) Primera estructura usada en la *heurística padre o no de la clase*; (b) Segunda estructura usada en la *heurística padre o no de la clase*.

nos da, es decir, el valor es mayor:

$$CMufInf_{f11}(X_i) = \sum_{j \neq i}^{n} MI(X_i, X_j|C).$$

**Filtro Bhattacharyya ($BMN_{f12}$)** Esta función [31] mide la dependencia que existe entre dos distribuciones de probabilidad. Las distribuciones que van a ser comparadas son las probabilidades a priori de una variable, contra la condicionada a la clase [15]. Intentamos de esta forma ver qué grado de dependencia encontramos entre ambas distribuciones; cuanto mayor sea ese grado, mayor será el peso de la variable analizada en el problema clasificatorio. Su formulación es:

$$Bh(X, C) = \sum_{i=1}^{m} -\log \left[ p(c_i) \sum_{j=1}^{k} \sqrt{p(x_j|c_i)p(x_j)} \right]$$

**Filtro ganancia-Dirichlet** Usamos la ganancia suavizada que vimos en el anterior capítulo. Utilizamos dos versiones en función del parámetro $s$. Probamos $s = 1$ ($BMN_{f13}$) y $s = 2$ ($BMN_{f14}$). En la función de ganancia las probabilidades son calculadas por máxima verosimilitud:

$$p_j = \frac{c_j}{m}$$

En este caso, las estimación de las probabilidades sigue una distribución de Dirichlet:

$$p_j^d = \frac{(s/n + c_j)}{(s + m)}$$

## 5.4. Resultados experimentales.

Se han seleccionado 25 conjuntos de datos, obtenidos del repositorio de aprendizaje automático *UCI repository of machine learning databases* [40], excepto *mofn-3-7-10* y *corral*, que fueron diseñados por Kohavi [197]. Todos estos conjuntos de datos han sido bastante utilizados en la literatura especializada para comparar clasificadores.

La tabla 6.2 muestra una breve descripción de las características de estas bases de datos: la columna *Instancias* muestra el número de casos, *Atributos* nos da el número de atributos usados para clasificar y, finalmente, la columna *Clases* muestra el número de estados distintos de la variable clase. Estos conjuntos de datos han sido preprocesados de la siguiente manera: los valores continuos se han discretizado, para ello se ha seguido el proceso propuesto por Fayyad e Irani [111]; los casos con valores perdidos han sido eliminados.

Para los experimentos se ha usado validación cruzada de 10-hojas para calcular la precisión de cada método, para cada uno de los problemas estudiados. Además se ha calculado el valor absoluto del logaritmo de la verosimilitud ($LogLikelihood = ln(P(c_i|\mathbf{x}))$), para obtener la calidad en la estimación de la probabilidad real de la clase. También se ha utilizado el test de Wilcoxon de signos pareados por rangos para determinar si los resultados de cada propuesta son estadísticamente significativos.

| # | Problema | Instancias | Atributos | Clases |
|---|---|---|---|---|
| 1 | australian | 690 | 14 | 2 |
| 2 | breast | 682 | 10 | 2 |
| 3 | chess | 3196 | 36 | 2 |
| 4 | cleve | 296 | 13 | 2 |
| 5 | corral | 128 | 6 | 2 |
| 6 | crx | 653 | 15 | 2 |
| 7 | diabetes | 768 | 8 | 2 |
| 8 | flare | 1066 | 10 | 2 |
| 9 | german | 1000 | 20 | 2 |
| 10 | glass | 214 | 9 | 7 |
| 11 | glass2 | 163 | 9 | 2 |
| 12 | heart | 270 | 13 | 2 |
| 13 | hepatitis | 80 | 19 | 2 |
| 14 | iris | 150 | 4 | 3 |
| 15 | letter | 20000 | 16 | 26 |
| 16 | lymphography | 148 | 18 | 4 |
| 17 | mofn-3-7-10 | 1324 | 10 | 2 |
| 18 | pima | 768 | 8 | 2 |
| 19 | satimage | 6435 | 36 | 6 |
| 20 | segment | 2310 | 19 | 7 |
| 21 | shuttle-small | 5800 | 9 | 7 |
| 22 | soybean-large | 562 | 35 | 19 |
| 23 | vehicle | 846 | 18 | 4 |
| 24 | vote | 435 | 16 | 2 |
| 25 | waveform-21 | 5000 | 21 | 3 |

Tabla 5.1: Descripción de los conjuntos de datos usados en los experimentos.

Nuestro objetivo es estudiar qué método es el más adecuado para construir nuestra multired en función de nuestras necesidades, para ello estudiaremos el método de envoltura expuesto en la sección 5.3.1 y los métodos de filtrado vistos en 5.3.2. En todas las multiredes aprendidas la red bayesiana utilizada en las hojas ha sido el clasificador naïve bayes. En las tablas 5.2, 5.3 y 5.4 tenemos el tanto por ciento de bien clasificados, junto con su desviación típica, para cada uno de los problemas y cada una de las multiredes propuestas. Además, utilizando los mismos cojuntos de validación, en la

tabla 5.5 tenemos los resultados para los clasificadores naïve bayes y C4.5, estos dos clasificadore se han incluido para estudiar la mejora introducidad por las multiredes. En la tabla 5.6 tenemos sólo la precisión de los distintos clasificadores, para poder ver todos los resultados en conjunto.

| Problema | $BMN_w$ | $BMN_{f1}$ | $BMN_{f2}$ | $BMN_{f3}$ | $BMN_{f4}$ |
|---|---|---|---|---|---|
| australian | $86,52 \pm 04,68$ | $86,38 \pm 05,23$ | $86,38 \pm 05,23$ | $86,38 \pm 05,23$ | $86,38 \pm 05,23$ |
| breast | $96,34 \pm 02,40$ | $95,89 \pm 01,99$ | $95,89 \pm 02,27$ | $95,89 \pm 02,27$ | $95,90 \pm 02,36$ |
| chess | $94,24 \pm 02,04$ | $93,52 \pm 01,99$ | $93,52 \pm 01,99$ | $93,52 \pm 01,99$ | $93,52 \pm 01,99$ |
| cleve | $82,45 \pm 07,29$ | $81,78 \pm 05,17$ | $82,12 \pm 05,37$ | $82,12 \pm 05,37$ | $82,09 \pm 05,80$ |
| corral | $88,33 \pm 07,03$ | $85,19 \pm 08,00$ | $85,19 \pm 08,00$ | $85,19 \pm 08,00$ | $85,19 \pm 08,00$ |
| crx | $86,97 \pm 05,09$ | $86,97 \pm 05,05$ | $86,97 \pm 05,05$ | $86,97 \pm 05,05$ | $86,97 \pm 05,05$ |
| diabetes | $77,48 \pm 04,06$ | $77,07 \pm 03,53$ | $77,07 \pm 03,53$ | $77,07 \pm 03,53$ | $77,07 \pm 03,53$ |
| flare | $81,70 \pm 02,98$ | $81,80 \pm 02,95$ | $81,80 \pm 02,95$ | $81,80 \pm 02,95$ | $78,89 \pm 04,19$ |
| german | $74,40 \pm 04,48$ | $75,20 \pm 04,42$ | $75,20 \pm 04,42$ | $75,20 \pm 04,42$ | $75,20 \pm 04,42$ |
| glass | $74,87 \pm 09,28$ | $71,06 \pm 07,04$ | $72,97 \pm 06,19$ | $72,99 \pm 10,52$ | $72,47 \pm 09,74$ |
| glass2 | $85,92 \pm 06,64$ | $84,71 \pm 06,19$ | $84,71 \pm 06,19$ | $84,71 \pm 06,19$ | $82,87 \pm 07,06$ |
| heart | $82,96 \pm 08,80$ | $82,96 \pm 06,87$ | $82,96 \pm 06,87$ | $82,96 \pm 06,87$ | $82,96 \pm 06,87$ |
| hepatitis | $88,75 \pm 08,75$ | $87,50 \pm 11,18$ | $87,50 \pm 13,69$ | $87,50 \pm 11,18$ | $87,50 \pm 11,18$ |
| iris | $94,00 \pm 02,29$ | $94,67 \pm 06,53$ | $94,00 \pm 06,29$ | $94,00 \pm 06,29$ | $94,00 \pm 06,29$ |
| letter | $83,56 \pm 00,58$ | $82,39 \pm 00,60$ | $82,39 \pm 00,60$ | $82,39 \pm 00,60$ | $78,93 \pm 00,64$ |
| lymphography | $84,48 \pm 14,34$ | $82,33 \pm 08,81$ | $79,67 \pm 12,77$ | $79,00 \pm 16,80$ | $80,38 \pm 15,40$ |
| mofn-3-7-10 | $94,34 \pm 01,95$ | $93,88 \pm 01,87$ | $93,88 \pm 01,87$ | $94,03 \pm 01,74$ | $94,03 \pm 01,74$ |
| pima | $77,34 \pm 06,08$ | $77,47 \pm 05,49$ | $77,47 \pm 05,49$ | $77,47 \pm 05,49$ | $77,47 \pm 05,49$ |
| satimage | $85,78 \pm 01,17$ | $84,27 \pm 01,47$ | $85,21 \pm 01,56$ | $84,77 \pm 01,55$ | $84,27 \pm 01,47$ |
| segment | $94,76 \pm 01,84$ | $93,81 \pm 01,06$ | $93,81 \pm 01,06$ | $93,81 \pm 01,06$ | $93,20 \pm 00,78$ |
| shuttle-small | $99,62 \pm 00,22$ | $99,17 \pm 00,34$ | $99,17 \pm 00,34$ | $99,17 \pm 00,34$ | $99,17 \pm 00,34$ |
| soybean-large | $91,46 \pm 04,27$ | $91,64 \pm 04,37$ | $85,58 \pm 03,36$ | $91,46 \pm 04,27$ | $91,46 \pm 04,56$ |
| vehicle | $71,17 \pm 04,64$ | $71,29 \pm 03,74$ | $71,29 \pm 03,74$ | $71,29 \pm 03,74$ | $67,39 \pm 05,12$ |
| vote | $94,28 \pm 03,69$ | $92,20 \pm 03,70$ | $92,20 \pm 03,70$ | $92,20 \pm 03,70$ | $92,20 \pm 03,70$ |
| waveform-21 | $82,94 \pm 00,91$ | $81,72 \pm 01,26$ | $81,72 \pm 01,26$ | $81,72 \pm 01,26$ | $81,52 \pm 01,24$ |

Tabla 5.2: Resultados Experimentales: precisión y su desviación estándar (parte 1 de 3).

| Problema | $BMN_{f5}$ | $BMN_{f6}$ | $BMN_{f7}$ | $BMN_{f8}$ | $BMN_{f9}$ |
|---|---|---|---|---|---|
| australian | $86,52 \pm 04,20$ | $86,52 \pm 04,20$ | $86,52 \pm 04,20$ | $86,38 \pm 05,23$ | $86,38 \pm 05,23$ |
| breast | $95,89 \pm 02,27$ | $96,60 \pm 02,38$ | $96,63 \pm 02,19$ | $95,89 \pm 02,27$ | $95,89 \pm 02,27$ |
| chess | $84,61 \pm 01,36$ | $84,61 \pm 01,36$ | $84,61 \pm 01,36$ | $93,52 \pm 01,99$ | $93,52 \pm 01,99$ |
| cleve | $82,44 \pm 06,13$ | $82,78 \pm 07,11$ | $82,78 \pm 07,11$ | $81,78 \pm 05,17$ | $81,78 \pm 05,17$ |
| corral | $87,56 \pm 06,97$ | $88,40 \pm 07,85$ | $87,56 \pm 06,97$ | $85,19 \pm 08,00$ | $85,19 \pm 08,00$ |
| crx | $86,97 \pm 04,85$ | $86,66 \pm 04,49$ | $86,97 \pm 04,85$ | $86,97 \pm 05,05$ | $86,97 \pm 05,05$ |
| diabetes | $77,61 \pm 03,56$ | $77,61 \pm 03,56$ | $77,61 \pm 03,56$ | $77,07 \pm 03,53$ | $77,07 \pm 03,53$ |
| flare | $81,80 \pm 02,95$ | $81,80 \pm 02,95$ | $81,80 \pm 02,95$ | $81,80 \pm 02,95$ | $81,80 \pm 02,95$ |
| german | $73,10 \pm 04,06$ | $75,00 \pm 03,74$ | $75,50 \pm 04,72$ | $75,20 \pm 04,42$ | $75,20 \pm 04,42$ |
| glass | $76,23 \pm 08,50$ | $73,42 \pm 09,35$ | $73,42 \pm 09,35$ | $74,85 \pm 09,78$ | $74,85 \pm 09,78$ |
| glass2 | $84,71 \pm 06,19$ | $84,12 \pm 07,15$ | $85,33 \pm 05,45$ | $84,71 \pm 06,19$ | $84,71 \pm 06,19$ |
| heart | $82,22 \pm 09,04$ | $83,70 \pm 07,80$ | $83,70 \pm 07,44$ | $82,96 \pm 06,87$ | $82,96 \pm 06,87$ |
| hepatitis | $90,00 \pm 09,35$ | $85,00 \pm 10,90$ | $85,00 \pm 10,90$ | $87,50 \pm 11,18$ | $87,50 \pm 11,18$ |
| iris | $94,00 \pm 06,29$ | $94,67 \pm 05,81$ | $93,33 \pm 05,96$ | $94,00 \pm 06,29$ | $94,00 \pm 06,29$ |
| letter | $83,56 \pm 00,59$ | $73,99 \pm 01,02$ | $79,91 \pm 00,73$ | $82,39 \pm 00,60$ | $82,40 \pm 00,61$ |
| lymphography | $83,10 \pm 10,88$ | $85,10 \pm 09,88$ | $84,48 \pm 10,57$ | $82,33 \pm 08,81$ | $82,33 \pm 08,81$ |
| mofn-3-7-10 | $94,79 \pm 02,08$ | $94,79 \pm 02,08$ | $94,79 \pm 02,08$ | $94,03 \pm 01,74$ | $94,03 \pm 01,74$ |
| pima | $77,73 \pm 05,55$ | $77,73 \pm 05,55$ | $77,73 \pm 05,55$ | $77,47 \pm 05,49$ | $77,47 \pm 05,49$ |
| satimage | $86,01 \pm 01,13$ | $85,50 \pm 00,87$ | $86,01 \pm 01,13$ | $84,27 \pm 01,47$ | $85,11 \pm 01,44$ |
| segment | $95,02 \pm 01,10$ | $92,08 \pm 01,85$ | $95,02 \pm 01,10$ | $93,81 \pm 01,06$ | $93,81 \pm 01,06$ |
| shuttle-small | $99,53 \pm 00,32$ | $99,02 \pm 00,37$ | $99,53 \pm 00,32$ | $99,17 \pm 00,34$ | $99,17 \pm 00,34$ |
| soybean-large | $91,99 \pm 03,83$ | $91,46 \pm 04,26$ | $92,17 \pm 04,15$ | $91,64 \pm 04,37$ | $91,64 \pm 04,36$ |
| vehicle | $70,82 \pm 03,63$ | $69,88 \pm 04,87$ | $70,47 \pm 04,41$ | $71,29 \pm 03,74$ | $70,70 \pm 04,07$ |
| vote | $93,12 \pm 03,04$ | $93,12 \pm 03,04$ | $93,12 \pm 03,04$ | $92,20 \pm 03,70$ | $92,20 \pm 03,70$ |
| waveform-21 | $81,52 \pm 01,24$ | $81,52 \pm 01,24$ | $81,52 \pm 01,24$ | $81,72 \pm 01,26$ | $81,72 \pm 01,26$ |

Tabla 5.3: Resultados Experimentales: precisión y su desviación estándar (parte 2 de 3).

| Problema | $BMN_{f10}$ | $BMN_{f11}$ | $BMN_{f12}$ | $BMN_{f13}$ | $BMN_{f14}$ |
|---|---|---|---|---|---|
| australian | $86,38 \pm 05,23$ | $84,49 \pm 03,24$ | $86,38 \pm 05,23$ | $86,38 \pm 05,23$ | $86,38 \pm 05,23$ |
| breast | $95,89 \pm 02,27$ | $95,89 \pm 02,27$ | $95,89 \pm 02,27$ | $95,89 \pm 02,27$ | $95,89 \pm 02,27$ |
| chess | $93,52 \pm 01,99$ | $84,61 \pm 01,37$ | $93,52 \pm 01,99$ | $93,52 \pm 01,99$ | $93,52 \pm 01,99$ |
| cleve | $81,78 \pm 05,17$ | $83,82 \pm 09,15$ | $82,12 \pm 05,37$ | $81,78 \pm 05,17$ | $81,78 \pm 05,17$ |
| corral | $85,19 \pm 08,00$ | $88,40 \pm 07,85$ | $85,19 \pm 08,00$ | $85,19 \pm 08,00$ | $85,19 \pm 08,00$ |
| crx | $86,97 \pm 05,05$ | $85,60 \pm 06,36$ | $86,97 \pm 05,05$ | $86,97 \pm 05,05$ | $86,97 \pm 05,05$ |
| diabetes | $77,07 \pm 03,53$ | $77,61 \pm 03,56$ | $77,07 \pm 03,53$ | $77,07 \pm 03,53$ | $77,07 \pm 03,53$ |
| flare | $81,80 \pm 02,95$ | $81,80 \pm 02,95$ | $81,80 \pm 02,95$ | $81,80 \pm 02,95$ | $81,80 \pm 02,95$ |
| german | $75,20 \pm 04,42$ | $73,50 \pm 03,61$ | $75,20 \pm 04,42$ | $75,20 \pm 04,42$ | $75,20 \pm 04,42$ |
| glass | $74,85 \pm 09,78$ | $72,97 \pm 06,19$ | $72,97 \pm 06,19$ | $73,46 \pm 11,77$ | $72,99 \pm 10,95$ |
| glass2 | $84,71 \pm 06,19$ | $84,12 \pm 07,15$ | $84,71 \pm 06,19$ | $84,71 \pm 06,19$ | $84,71 \pm 06,19$ |
| heart | $82,96 \pm 06,87$ | $83,33 \pm 07,08$ | $82,96 \pm 06,87$ | $82,96 \pm 06,87$ | $82,96 \pm 06,87$ |
| hepatitis | $87,50 \pm 11,18$ | $88,75 \pm 10,38$ | $87,50 \pm 13,69$ | $87,50 \pm 11,18$ | $88,75 \pm 10,38$ |
| iris | $94,00 \pm 06,29$ | $94,67 \pm 05,81$ | $94,00 \pm 06,29$ | $94,00 \pm 06,29$ | $94,00 \pm 06,29$ |
| letter | $82,39 \pm 00,60$ | $83,56 \pm 00,59$ | $82,30 \pm 00,60$ | $82,39 \pm 00,60$ | $82,40 \pm 00,61$ |
| lymphography | $82,33 \pm 08,81$ | $77,81 \pm 12,24$ | $83,05 \pm 16,23$ | $82,33 \pm 08,81$ | $82,33 \pm 08,81$ |
| mofn-3-7-10 | $94,03 \pm 01,74$ | $94,71 \pm 02,06$ | $93,88 \pm 01,87$ | $94,03 \pm 01,74$ | $94,03 \pm 01,74$ |
| pima | $77,47 \pm 05,49$ | $77,73 \pm 05,55$ | $77,47 \pm 05,49$ | $77,47 \pm 05,49$ | $77,47 \pm 05,49$ |
| satimage | $84,27 \pm 01,47$ | $86,01 \pm 01,13$ | $84,27 \pm 01,47$ | $84,77 \pm 01,55$ | $84,77 \pm 01,55$ |
| segment | $93,81 \pm 01,06$ | $95,02 \pm 01,10$ | $93,81 \pm 01,06$ | $93,81 \pm 01,06$ | $93,81 \pm 01,06$ |
| shuttle-small | $99,17 \pm 00,34$ | $99,17 \pm 00,34$ | $99,17 \pm 00,34$ | $99,17 \pm 00,34$ | $99,17 \pm 00,34$ |
| soybean-large | $91,64 \pm 04,37$ | $85,58 \pm 03,36$ | $85,58 \pm 03,36$ | $91,64 \pm 04,37$ | $91,64 \pm 04,37$ |
| vehicle | $70,23 \pm 03,68$ | $71,29 \pm 03,74$ | $71,29 \pm 03,74$ | $71,29 \pm 03,74$ | $71,29 \pm 03,74$ |
| vote | $92,20 \pm 03,70$ | $93,82 \pm 03,20$ | $92,20 \pm 03,70$ | $92,20 \pm 03,70$ | $92,20 \pm 03,70$ |
| waveform-21 | $81,72 \pm 01,26$ | $81,72 \pm 01,26$ | $81,72 \pm 01,26$ | $81,72 \pm 01,26$ | $81,72 \pm 01,26$ |

Tabla 5.4: Resultados Experimentales: precisión y su desviación estándar (parte 3 de 3).

| Problema | naïve bayes | C4.5 |
|---|---|---|
| australian | $85,22 \pm 03,71$ | $85,51 \pm 04,85$ |
| breast | $97,66 \pm 01,76$ | $91,96 \pm 03,33$ |
| chess | $87,89 \pm 01,17$ | $94,34 \pm 02,02$ |
| cleve | $82,78 \pm 07,11$ | $73,55 \pm 09,09$ |
| corral | $86,74 \pm 06,93$ | $76,60 \pm 11,34$ |
| crx | $86,81 \pm 04,64$ | $86,36 \pm 04,33$ |
| diabetes | $78,12 \pm 03,52$ | $75,36 \pm 05,27$ |
| flare | $80,49 \pm 04,47$ | $82,64 \pm 02,68$ |
| german | $75,50 \pm 04,72$ | $70,00 \pm 05,12$ |
| glass | $73,42 \pm 09,35$ | $47,94 \pm 12,22$ |
| glass2 | $84,08 \pm 07,16$ | $59,96 \pm 14,18$ |
| heart | $83,70 \pm 07,44$ | $72,59 \pm 09,10$ |
| hepatitis | $85,00 \pm 10,90$ | $78,75 \pm 13,75$ |
| iris | $94,00 \pm 04,67$ | $94,67 \pm 06,53$ |
| letter | $73,99 \pm 01,02$ | $76,75 \pm 01,02$ |
| lymphography | $83,05 \pm 14,19$ | $79,71 \pm 10,83$ |
| mofn-3-7-10 | $85,50 \pm 03,56$ | $77,95 \pm 03,37$ |
| pima | $78,11 \pm 06,35$ | $77,59 \pm 05,22$ |
| satimage | $82,47 \pm 01,15$ | $82,66 \pm 01,10$ |
| segment | $92,08 \pm 01,85$ | $94,16 \pm 00,89$ |
| shuttle-small | $99,02 \pm 00,37$ | $99,54 \pm 00,27$ |
| soybean-large | $91,46 \pm 04,56$ | $91,11 \pm 03,82$ |
| vehicle | $63,36 \pm 03,18$ | $63,13 \pm 04,03$ |
| vote | $90,13 \pm 03,39$ | $94,27 \pm 02,08$ |
| waveform-21 | $81,72 \pm 01,22$ | $74,62 \pm 02,16$ |

Tabla 5.5: Resultados Experimentales: precisión y desviación estándar para los clasificadores naïve bayes y C4.5.

| # | $w$ | $f1$ | $f2$ | $f3$ | $f4$ | $f5$ | $f6$ | $f7$ | $f8$ | $f9$ | $f10$ | $f11$ | $f12$ | $f12$ | $f14$ | $NB$ | $C4{,}5$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 86,52 | 86,38 | 86,38 | 86,38 | 86,38 | 86,52 | 86,52 | 86,52 | 86,38 | 86,38 | 86,38 | 84,49 | 86,38 | 86,38 | 86,38 | 85,22 | 85,51 |
| 2 | 96,34 | 95,89 | 95,89 | 95,89 | 95,90 | 95,89 | 96,60 | 96,63 | 95,89 | 95,89 | 95,89 | 95,89 | 95,89 | 95,89 | 95,89 | 97,66 | 91,96 |
| 3 | 94,24 | 93,52 | 93,52 | 93,52 | 93,52 | 84,61 | 84,61 | 84,61 | 93,52 | 93,52 | 93,52 | 84,61 | 93,52 | 93,52 | 93,52 | 87,89 | 94,34 |
| 4 | 82,45 | 81,78 | 82,12 | 82,12 | 82,09 | 82,44 | 82,78 | 82,78 | 81,78 | 81,78 | 81,78 | 83,82 | 82,12 | 81,78 | 81,78 | 82,78 | 73,55 |
| 5 | 88,33 | 85,19 | 85,19 | 85,19 | 85,19 | 87,56 | 88,40 | 87,56 | 85,19 | 85,19 | 85,19 | 88,40 | 85,19 | 85,19 | 85,19 | 86,74 | 76,60 |
| 6 | 86,97 | 86,97 | 86,97 | 86,97 | 86,97 | 86,97 | 86,66 | 86,97 | 86,97 | 86,97 | 86,97 | 85,60 | 86,97 | 86,97 | 86,97 | 86,81 | 86,36 |
| 7 | 77,48 | 77,07 | 77,07 | 77,07 | 77,07 | 77,61 | 77,61 | 77,61 | 77,07 | 77,07 | 77,07 | 77,61 | 77,07 | 77,07 | 77,07 | 78,12 | 75,36 |
| 8 | 81,70 | 81,80 | 81,80 | 81,80 | 78,89 | 81,80 | 81,80 | 81,80 | 81,80 | 81,80 | 81,80 | 81,80 | 81,80 | 81,80 | 81,80 | 80,49 | 82,64 |
| 9 | 74,40 | 75,20 | 75,20 | 75,20 | 75,20 | 73,10 | 75,00 | 75,50 | 75,20 | 75,20 | 75,20 | 73,50 | 75,20 | 75,20 | 75,20 | 75,50 | 70,00 |
| 10 | 74,87 | 71,06 | 72,97 | 72,99 | 72,47 | 76,23 | 73,42 | 73,42 | 74,85 | 74,85 | 74,85 | 72,97 | 72,97 | 73,46 | 72,99 | 73,42 | 47,94 |
| 11 | 85,92 | 84,71 | 84,71 | 84,71 | 82,87 | 84,71 | 84,12 | 85,33 | 84,71 | 84,71 | 84,71 | 84,12 | 84,71 | 84,71 | 84,71 | 84,08 | 59,96 |
| 12 | 82,96 | 82,96 | 82,96 | 82,96 | 82,96 | 82,22 | 83,70 | 83,70 | 82,96 | 82,96 | 82,96 | 83,33 | 82,96 | 82,96 | 82,96 | 83,70 | 72,59 |
| 13 | 88,75 | 87,50 | 87,50 | 87,50 | 87,50 | 90,00 | 85,00 | 85,00 | 87,50 | 87,50 | 87,50 | 88,75 | 87,50 | 87,50 | 88,75 | 85,00 | 78,75 |
| 14 | 94,00 | 94,67 | 94,00 | 94,00 | 94,00 | 94,00 | 94,67 | 93,33 | 94,00 | 94,00 | 94,00 | 94,67 | 94,00 | 94,00 | 94,00 | 94,00 | 94,67 |
| 15 | 83,56 | 82,39 | 82,40 | 82,39 | 78,93 | 83,57 | 73,99 | 79,91 | 82,39 | 82,40 | 82,39 | 83,56 | 82,30 | 82,39 | 82,40 | 73,99 | 76,75 |
| 16 | 84,48 | 82,33 | 79,67 | 79,00 | 80,38 | 83,10 | 85,10 | 84,48 | 82,33 | 82,33 | 82,33 | 77,81 | 83,05 | 82,33 | 82,33 | 83,05 | 79,71 |
| 17 | 94,34 | 93,88 | 93,88 | 94,03 | 94,03 | 94,79 | 94,79 | 94,79 | 94,03 | 94,03 | 94,03 | 94,71 | 93,88 | 94,03 | 94,03 | 85,50 | 77,95 |
| 18 | 77,34 | 77,47 | 77,47 | 77,47 | 77,47 | 77,73 | 77,73 | 77,73 | 77,47 | 77,47 | 77,47 | 77,73 | 77,47 | 77,47 | 77,47 | 78,11 | 77,59 |
| 19 | 85,78 | 84,27 | 85,21 | 84,77 | 84,27 | 86,01 | 85,50 | 86,01 | 84,27 | 85,11 | 84,27 | 86,01 | 84,27 | 84,77 | 84,77 | 82,47 | 82,66 |
| 20 | 94,76 | 93,81 | 93,81 | 93,81 | 93,20 | 95,02 | 92,08 | 95,02 | 93,81 | 93,81 | 93,81 | 95,02 | 93,81 | 93,81 | 93,81 | 92,08 | 94,16 |
| 21 | 99,62 | 99,17 | 99,17 | 99,17 | 99,17 | 99,53 | 99,02 | 99,53 | 99,17 | 99,17 | 99,17 | 99,17 | 99,17 | 99,17 | 99,17 | 99,02 | 99,54 |
| 22 | 91,46 | 91,64 | 85,58 | 91,46 | 91,46 | 91,81 | 91,46 | 92,00 | 91,64 | 91,64 | 91,64 | 85,58 | 85,58 | 91,64 | 91,64 | 91,46 | 91,11 |
| 23 | 71,17 | 71,29 | 71,29 | 71,29 | 67,39 | 70,82 | 69,88 | 70,47 | 71,29 | 70,70 | 70,23 | 71,29 | 71,29 | 71,29 | 71,29 | 63,36 | 63,13 |
| 24 | 94,28 | 92,20 | 92,20 | 92,20 | 92,20 | 93,12 | 93,12 | 93,12 | 92,20 | 92,20 | 92,20 | 93,82 | 92,20 | 92,20 | 92,20 | 90,13 | 94,27 |
| 25 | 82,94 | 81,72 | 81,72 | 81,72 | 81,52 | 81,52 | 81,52 | 81,52 | 81,72 | 81,72 | 81,72 | 81,72 | 81,72 | 81,72 | 81,72 | 81,72 | 74,62 |
| $\bar{\mathbf{x}}$ | 86,19 | 85,40 | 85,15 | 85,35 | 84,84 | 85,63 | 84,96 | 85,42 | 85,53 | 85,54 | 85,48 | 85,04 | 85,24 | 85,49 | 85,52 | 84,09 | 80,07 |

Tabla 5.6: Resultados experimentales: precisión de cada propuesta para cada problema.

Observando estos resultados vemos que el método de envoltura (la columna $w$ en la tabla 5.6 ) obtiene mejores resultados, en general, que el resto de los métodos de filtrado. Especialmente parece mejor el método de envoltura en problemas con un gran número atributos como son *chess*, *waveform-21*, *letter* o *satimage*. Esto es debido a la búsqueda más exhaustiva y orientada a la precisión del clasificador, realizada en el método de envoltura, haciéndose más patente en estos problemas donde existe una gran numero de variables. Téngase en cuenta también que cuantas más variables tiene el problema más tiempo consume la técnica de envoltura en comparación con las de filtrado.

En la tabla 5.7 se muestra el opuesto del logaritmo de la verosimilitud para las distintas multiredes y problemas, valores más pequeños indican mejores resultados. En esta tabla observamos que aunque el método de envoltura obtiene mejores resultados en el tanto por ciento de bien clasificados estas predicciones no siempre son más ajustadas que los métodos de filtrado.

| # | $w$ | $f1$ | $f2$ | $f3$ | $f4$ | $f5$ | $f6$ | $f7$ | $f8$ | $f9$ | $f10$ | $f11$ | $f12$ | $f12$ | $f14$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 10,00 | 10,40 | 10,40 | 10,40 | 10,40 | 09,86 | 09,86 | 09,86 | 10,40 | 10,40 | 10,40 | 08,56 | 10,40 | 10,40 | 10,40 |
| 2 | 10,47 | 09,88 | 09,88 | 09,88 | 10,36 | 09,88 | 10,41 | 11,09 | 09,88 | 09,88 | 09,88 | 09,88 | 09,88 | 09,88 | 09,88 |
| 3 | 14,05 | 14,19 | 14,19 | 14,19 | 14,19 | 13,30 | 13,30 | 13,30 | 14,19 | 14,19 | 14,19 | 13,30 | 14,19 | 14,19 | 14,19 |
| 4 | 07,29 | 07,07 | 07,04 | 07,04 | 07,34 | 07,35 | 07,97 | 07,97 | 07,07 | 07,11 | 07,11 | 06,94 | 07,04 | 07,07 | 07,07 |
| 5 | 03,76 | 03,93 | 03,93 | 03,93 | 03,93 | 03,79 | 03,79 | 03,79 | 03,93 | 03,93 | 03,93 | 03,79 | 03,93 | 03,93 | 03,93 |
| 6 | 10,62 | 10,57 | 10,57 | 10,57 | 10,57 | 10,22 | 10,40 | 10,22 | 10,57 | 10,57 | 10,57 | 08,76 | 10,57 | 10,57 | 10,57 |
| 7 | 04,37 | 03,81 | 03,81 | 03,81 | 03,81 | 04,35 | 04,35 | 04,35 | 03,81 | 03,81 | 03,81 | 04,35 | 03,81 | 03,81 | 03,81 |
| 8 | 03,56 | 03,09 | 03,09 | 03,09 | 05,92 | 03,09 | 03,09 | 03,09 | 03,09 | 03,09 | 03,16 | 03,09 | 03,09 | 03,09 | 03,09 |
| 9 | 14,26 | 13,43 | 13,43 | 13,43 | 13,43 | 13,72 | 14,04 | 14,67 | 13,43 | 13,43 | 13,42 | 12,74 | 13,43 | 13,43 | 13,43 |
| 10 | 03,91 | 03,68 | 03,53 | 03,68 | 04,22 | 03,98 | 04,64 | 04,64 | 03,87 | 03,87 | 03,87 | 03,53 | 03,53 | 03,72 | 03,81 |
| 11 | 02,86 | 02,42 | 02,28 | 02,28 | 02,90 | 02,28 | 02,28 | 02,27 | 02,42 | 02,42 | 02,42 | 02,30 | 02,28 | 02,42 | 02,42 |
| 12 | 05,52 | 05,19 | 05,19 | 05,19 | 05,19 | 05,17 | 05,67 | 05,90 | 05,19 | 05,19 | 05,19 | 05,17 | 05,18 | 05,19 | 05,19 |
| 13 | 07,86 | 07,99 | 07,87 | 07,99 | 07,99 | 07,84 | 08,53 | 08,53 | 07,99 | 07,99 | 07,99 | 07,63 | 07,94 | 07,99 | 07,93 |
| 14 | 01,93 | 01,89 | 01,89 | 01,89 | 01,89 | 01,89 | 01,96 | 01,88 | 01,89 | 01,89 | 01,89 | 01,96 | 01,89 | 01,89 | 01,89 |
| 15 | 18,70 | 19,37 | 19,37 | 19,37 | 20,53 | 18,70 | 23,57 | 19,92 | 19,37 | 19,37 | 19,37 | 18,70 | 19,37 | 19,37 | 19,37 |
| 16 | 13,75 | 13,74 | 13,80 | 13,79 | 14,53 | 13,88 | 14,04 | 14,39 | 13,74 | 13,74 | 13,74 | 13,66 | 13,90 | 13,74 | 13,74 |
| 17 | 06,49 | 06,50 | 06,50 | 06,50 | 06,50 | 06,50 | 06,50 | 06,50 | 06,50 | 06,50 | 06,50 | 06,49 | 06,50 | 06,50 | 06,50 |
| 18 | 04,56 | 03,80 | 03,80 | 03,80 | 03,80 | 04,34 | 04,34 | 04,34 | 03,80 | 03,80 | 03,80 | 04,34 | 03,80 | 03,80 | 03,80 |
| 19 | 44,04 | 45,06 | 45,29 | 45,07 | 45,06 | 43,85 | 44,90 | 43,85 | 45,06 | 45,18 | 45,06 | 43,85 | 45,06 | 45,07 | 45,07 |
| 20 | 14,59 | 15,51 | 15,51 | 15,51 | 16,52 | 14,12 | 20,29 | 14,12 | 15,51 | 15,51 | 15,51 | 14,12 | 15,51 | 15,51 | 15,51 |
| 21 | 06,03 | 06,84 | 06,84 | 06,84 | 06,84 | 05,69 | 07,94 | 05,69 | 06,84 | 06,84 | 06,84 | 06,84 | 06,84 | 06,84 | 06,84 |
| 22 | 15,16 | 15,33 | 16,98 | 14,90 | 15,58 | 14,74 | 15,72 | 14,64 | 14,95 | 14,95 | 14,95 | 16,98 | 16,98 | 14,95 | 14,95 |
| 23 | 10,56 | 09,01 | 09,01 | 09,01 | 10,27 | 09,05 | 09,10 | 09,08 | 09,01 | 09,09 | 09,10 | 09,01 | 09,01 | 09,01 | 09,01 |
| 24 | 03,69 | 10,08 | 10,08 | 10,08 | 10,08 | 09,60 | 09,61 | 09,60 | 10,08 | 10,08 | 10,08 | 09,61 | 10,08 | 10,08 | 10,08 |
| 25 | 25,02 | 24,88 | 24,88 | 24,88 | 25,02 | 25,02 | 25,02 | 25,02 | 24,88 | 24,88 | 24,88 | 24,88 | 24,88 | 24,88 | 24,88 |
| $\bar{\mathbf{x}}$ | 10,77 | 10,71 | 10,77 | 10,68 | 11,07 | 10,49 | 11,25 | 10,75 | 10,70 | 10,71 | 10,71 | 10,42 | 10,76 | 10,69 | 10,69 |

Tabla 5.7: Resultados experimentales: logaritmo de la verosimilitud.

En la tabla 5.8 podemos ver un resumen, donde se compara cada clasificador con los otros. La entrada en la fila $i$ y columna $j$, representa el número de veces que el clasificador $i$ es mejor/significativamente mejor (usando el test de Wilcoxon) que el clasificador $j$. Cada fila muestra las veces que el clasificador correspondiente es mejor, considerando que cada columna dice cuantas veces cada clasificador es peor.

Estos resultados indican claramente que el método de envoltura generalmente tiene un comportamiento mejor que el resto de propuestas, especialmente cuando nos centramos en diferencias estadísticamente significativas. Siguiendo al método de envoltura, tenemos la heurística *atributo padre del resto* con la métrica K2 ($BMN_{f5}$). Esta heurística es mejor que la búsqueda de envoltura en 13 casos, mientras que la búsqueda de envoltura es sólo mejor en 8 casos, aunque sólo dos de ellos son significativos estadísticamente. Con un comportamiento similar, pero un poco peor que la medida $BMN_{f5}$, encontramos la heurística *Atributo padre del resto* con métrica BDe ($BMN_{f7}$), *Padre o no de la clase* con métrica BIC ($BMN_{f9}$) e *información mutua condicionada* ($BMN_{f11}$). También queremos destacar los buenos resultados obtenidos por el *filtro Bhattacharyya* ($BMN_{f12}$), pero peores que los anteriores métodos de filtrado mencionados y peor que la búsqueda de envoltura.

Otra conclusión que podemos obtener es que las multiredes funcionan mejor que el clasificador naïve bayes (en ningún problema éste es significativamente mejor) y que el clasificador C4.5.

| | $w$ | $f1$ | $f2$ | $f3$ | $f4$ | $f5$ | $f6$ | $f7$ | $f8$ | $f9$ | $f10$ | $f11$ | $f12$ | $f13$ | $f14$ | $NB$ | $CT$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| w | — | 18/**6** | 19/**5** | 18/**5** | 20/**9** | 12/**3** | 13/**5** | 11/**3** | 18/**5** | 19/**4** | 19/**5** | 12/**5** | 19/**5** | 18/**5** | 17/**7** | 17/**8** | 21/**12** |
| f1 | 6/**0** | — | 3/**1** | 3/**0** | 9/**3** | 6/**1** | 12/**3** | 6/**2** | 1/**0** | 2/**0** | 2/**1** | 7/**2** | 2/**1** | 1/**0** | 1/**0** | 15/**7** | 18/**12** |
| f2 | 4/**0** | 3/**1** | — | 2/**0** | 9/**4** | 5/**1** | 11/**3** | 6/**2** | 2/**1** | 3/**0** | 3/**2** | 6/**1** | 1/**1** | 2/**0** | 2/**0** | 13/**3** | 16/**1** |
| f3 | 4/**0** | 4/**0** | 3/**1** | — | 9/**3** | 5/**1** | 11/**3** | 6/**2** | 2/**0** | 2/**0** | 3/**1** | 8/**2** | 4/**1** | 1/**0** | 1/**0** | 13/**7** | 17/**12** |
| f4 | 2/**0** | 4/**0** | 4/**1** | 2/**0** | — | 4/**1** | 8/**2** | 3/**1** | 2/**0** | 2/**0** | 2/**0** | 7/**2** | 3/**1** | 2/**0** | 2/**0** | 11/**6** | 18/**12** |
| f5 | 10/**0** | 16/**5** | 16/**4** | 16/**3** | 19/**5** | — | 12/**4** | 5/**2** | 16/**4** | 17/**3** | 17/**4** | 9/**3** | 16/**6** | 16/**3** | 16/**3** | 16/**8** | 20/**14** |
| f6 | 11/**0** | 11/**0** | 13/**1** | 13/**0** | 16/**1** | 6/**1** | — | 3/**0** | 11/**0** | 11/**0** | 11/**0** | 8/**2** | 13/**2** | 11/**0** | 12/**0** | 11/**3** | 18/**11** |
| f7 | 12/**0** | 18/**3** | 18/**3** | 18/**2** | 21/**5** | 7/**1** | 11/**3** | — | 17/**3** | 17/**2** | 18/**3** | 11/**4** | 18/**5** | 17/**2** | 18/**2** | 14/**7** | 20/**13** |
| f8 | 5/**0** | 2/**0** | 4/**1** | 3/**0** | 9/**2** | 5/**1** | 13/**3** | 7/**2** | — | 1/**0** | 1/**1** | 8/**2** | 3/**1** | 1/**0** | 1/**0** | 15/**7** | 18/**12** |
| f9 | 4/**0** | 3/**0** | 4/**1** | 4/**0** | 10/**2** | 4/**1** | 13/**3** | 7/**2** | 1/**0** | — | 2/**1** | 8/**2** | 4/**2** | 2/**0** | 2/**0** | 15/**7** | 18/**12** |
| f10 | 4/**0** | 2/**0** | 4/**1** | 3/**0** | 9/**1** | 4/**1** | 13/**3** | 6/**2** | 0/**0** | 0/**0** | — | 8/**2** | 3/**1** | 1/**0** | 1/**0** | 15/**7** | 18/**12** |
| f11 | 11/**0** | 12/**3** | 12/**2** | 12/**2** | 17/**4** | 8/**0** | 10/**2** | 8/**1** | 12/**3** | 13/**2** | 13/**3** | — | 12/**3** | 12/**2** | 11/**2** | | |
| f12 | 4/**0** | 3/**0** | 1/**0** | 1/**0** | 9/**2** | 5/**1** | 11/**3** | 6/**2** | 2/**0** | 3/**0** | 3/**0** | 6/**1** | — | 2/**0** | 2/**0** | 13/**7** | 17/**12** |
| f13 | 5/**0** | 3/**0** | 4/**1** | 3/**0** | 10/**2** | 5/**1** | 13/**3** | 7/**2** | 1/**0** | 1/**0** | 2/**0** | 8/**2** | 4/**1** | — | 1/**0** | 15/**7** | 18/**12** |
| f14 | 5/**2** | 4/**0** | 5/**1** | 3/**0** | 11/**2** | 5/**1** | 12/**3** | 6/**2** | 2/**0** | 2/**0** | 3/**0** | 8/**2** | 5/**1** | 1/**0** | — | 14/**7** | 18/**12** |
| NB | 6/**0** | 9/**0** | 10/**0** | 9/**0** | 12/**0** | 8/**0** | 7/**0** | 6/**0** | 8/**0** | 8/**0** | 8/**0** | 11/**0** | 9/**0** | 8/**0** | 9/**0** | — | 16/**9** |
| CT | 4/**1** | 6/**4** | 9/**4** | 8/**4** | 7/**4** | 4/**2** | 6/**5** | 4/**2** | 7/**4** | 7/**4** | 7/**4** | 8/**3** | 8/**4** | 7/**4** | 7/**4** | 9/**5** | — |

Tabla 5.8: Número de veces que el clasificador de la fila $i$ es mejor/**significativamente mejor** que el clasificador de la columna $j$.

| | $BMN_w$ | $BMN_{f1}$ | $BMN_{f2}$ | $BMN_{f3}$ | $BMN_{f4}$ | $BMN_{f5}$ | $BMN_{f6}$ | $BMN_{f7}$ |
|---|---|---|---|---|---|---|---|---|
| $\bar{x}$ | 1 | 140,63 | 130,78 | 80,58 | 80,38 | 7,838 | 21,988 | 7,728 |

| | $BMN_{f8}$ | $BMN_{f9}$ | $BMN_{f10}$ | $BMN_{f11}$ | $BMN_{f12}$ | $BMN_{f13}$ | $BMN_{f14}$ |
|---|---|---|---|---|---|---|---|
| $\bar{x}$ | 65,558 | 99,18 | 65,688 | 4,588 | 142,278 | 79,118 | 79,668 |

Tabla 5.9: Resultados experimentales: tiempos de ejecución para cada multired en función del enfoque de envoltura.

En cuanto al tiempo consumido por cada propuesta, en la tabla 5.9 tenemos los tiempos para cada multired en función del tiempo consumido por el enfoque de envoltura (es decir, $\frac{t_{BMN_w}}{t_{BMN_f}}$). Como se esperaba, el método de envoltura es más lento que los métodos de filtrado. Según estos resultados podríamos hacer tres clasificaciones de los métodos, una donde se encuentren los algoritmos lentos (envoltura, solamente), otra donde estén los que tienen un consumo medio de tiempo ($BMN_{f5}$, $BMN_{f6}$, $BMN_{f7}$ y $BMN_{f11}$) y un tercer grupo, donde tenemos las propuestas más rápidas (el resto de los métodos de filtrado).

Atendiendo a la anterior clasificación temporal observamos que también dentro de esos grupos se pueden dividir las técnicas en función de su tanto por ciento de bien clasificados. Así, las técnicas lentas (envoltura) obtienen mejores resultados, seguidos por el segundo grupo que obtienen resultados muy próximos, aunque ligeramente peores, y en menos tiempo; finalmente en el grupo de las funciones rápidas tenemos peores resultados en tanto por ciento de bien clasificados, si bien el tiempo consumido es bastante menor. Del segundo grupo destacamos que el filtro $BMN_{f5}$ obtiene un rendimiento muy similar al método de envoltura en tanto por ciento de bien clasificados si bien es casi 8 veces mas rápido. Dentro del tercer grupo destacamos que la propuesta $BMN_{f12}$ obtiene buenos resultados siendo más de 142 veces más rápida que la técnica de envoltura.

Resumiendo, podríamos escoger la multired de envoltura para aquellos casos en los que el problema sea asequible en tiempo. Para aquellos problemas en los que la rapidez de cómputo sea fundamental, podemos utilizar la propuesta $BMN_{f12}$. Mientras que en problemas donde queramos llegar a un término medio podemos usar la heurística $BMN_{f5}$.

## 5.5. Discusión.

Las multiredes bayesianas son una extensión de las redes bayesianas ya que nos permiten representar independencias condicionadas al contexto. En este capítulo se ha realizado un profundo estudio de diferentes métodos para obtener la variable distinguida en las multiredes bayesianas.

De los métodos vistos, el método de envoltura es mejor que los de filtrado en cuanto a precisión en la clasificación, algo lógico si se tiene en cuenta que en el método de envoltura la medida que se busca maximizar es el tanto por ciento de bien clasificados. Aunque hemos encontrados métodos de filtrado que obtienen resultados muy similares al de envoltura en menos tiempo.

Lo métodos de filtrado también son mucho más rápidos en tiempo de ejecución, lo que los hace apropiados para problemas de gran tamaño o un gran número de variables, donde utilizar el esquema de envoltura sea impracticable. Además debemos tener en cuenta que los métodos de filtrado al ser independientes de los clasificadores bayesianos utilizados en las hojas pueden combinarse con clasificadores más potentes que el naïve bayes y esto, aunque se puede hacer en un método de envoltura, no tendría sentido.

Finalmente, como conclusión, hemos destacado tres métodos de obtención de la variable distinguida ($BMN_w$, $BMN_{f5}$ y $BMN_{f12}$) en función de las necesidades que tengamos.

# Capítulo 6

# Clasificador C-RPDAG: búsqueda en el espacio de grafos acíclicos parcialmente dirigidos.

El clasificador naïve bayes, pese a su simplicidad, ha tenido bastante aceptación pues muestra una sorprendente eficacia en el porcentaje de aciertos, como vimos en el primer capítulo. Otros clasificadores bayesianos han intentado aprovechar su éxito, y mejorarlo mediante la adición de arcos aumentados, que son enlaces entre atributos. Aquí tenemos, por ejemplo, los clasificadores TAN, KDB o BAN. No obstante, estas propuestas con arcos aumentados, conservan la estructura del naïve bayes, esto es, la variable clase es padre de los atributos. Hay otras aproximaciones que no usan arcos aumentados, como el clasificador semi naïve bayes que hace productos cartesianos de atributos, o el clasificador naïve bayes selectivo, que hace una selección de características. Pero estos clasificadores bayesianos, tienen la misma restricción estructural heredada del naïve bayes.

Las redes bayesianas (sin ningún tipo de restricción estructural) también pueden ser usadas en tareas de clasificación. En estos casos, cuando se usan como clasificadores, se les denomina *redes bayesianas sin restricciones* o *redes bayesianas generales*. Como vimos, cualquier red bayesiana puede ser utilizada en clasificación supervisada, para ello sólamente nos hace falta el manto de Markov de la variable clase. Al no tener ningún tipo de restricción

estructural, una red bayesiana general tiene más capacidad expresiva que el resto de clasificadores bayesianos. Entre otras, puede representar todas las estructuras de los clasificadores bayesianos que sí contienen forzosamente la estructura del naïve bayes.

No obstante, las redes bayesianas generales cuando son generadas por un enfoque de métrica+búsqueda, tratan de encontrar la red que maximiza el valor de la métrica y no la red que mejor clasifique [118], es por esto, que se les ha considerado malos clasificadores bayesianos. No obstante, en [69] se demostró que una red bayesiana construida mediante un algoritmo basado en tests de independencias obtenía mejores resultados.

Teniendo en mente la idea de que un clasificador bayesiano sin restricciones tiene mayor poder expresivo que un modelo restringido estructuralmente, nos vamos a centrar en este tipo de clasificadores bayesianos. En este capítulo se va a presentar un algoritmo para obtener un clasificador bayesiano sin restricciones, siguiendo un enfoque de métrica+búsqueda y que logra excelentes resultados.

El método realiza una simple búsqueda local, pero orientada hacia clasificación supervisada. Usará un espacio de búsqueda especializado, consistente en un tipo de grafo acíclico parcialmente dirigido, al que denominamos C-RPDAG y que combina dos conceptos de equivalencia de grafos acíclicos dirigidos: equivalencia en clasificación y equivalencia en independencia. Esto nos permite, por un lado, centrar la búsqueda desde la perspectiva de la clasificación y, por otro lado, explorar un espacio de búsqueda más robusto y reducido.

## 6.1. El espacio de búsqueda C-RPDAG.

En esta sección, vamos a describir los elementos que componen el espacio de búsqueda del algoritmo que se va a presentar para el aprendizaje de redes bayesianas como clasificadores.

En este contexto, el conjunto de variables que vamos a considerar será $\mathbf{X} = \{X_1, X_2, \ldots, X_n, C\}$. Donde $C$ es la variable de interés, que denominamos clase, y las variables $X_1, X_2, \ldots, X_n$ tienen el papel de atributos que serán utilizados para predecir el valor de $C$. El objetivo es calcular distribución a posteriori de la clase dada cualquier configuración de los atributos, $p(C|\mathbf{X} \setminus C)$.

### 6.1.1. Grafos dirigidos acíclicos centrados en la clase.

El primer hecho importante que debe tenerse en cuenta al diseñar nuestro espacio de búsqueda, es que desde el punto de vista de la clasificación, diferentes grafos dirigidos acíclicos (por abreviar, DAGs, del inglés *Directed Acyclic Graphs*) son equivalentes, en el sentido de que generan la misma distribución de probabilidad a posteriori para la variable clase. Hay que indicar que no son equivalentes en el sentido habitual, en las independencias o en la equivalencia de la distribución. Ampliaremos los detalles en la siguiente sección. Vamos a formalizar esta idea:

**Equivalencia en clasificación.** Sean $G = (\mathbf{X}, E_G)$ y $G' = (\mathbf{X}, E_{G'})$ dos grafos dirigidos acíclicos. Sea $D = \{\mathbf{u}^1, \ldots, \mathbf{u}^m\}$ un conjunto de datos con $m$ instancias. Sea $p$ cualquier distribución de probabilidad sobre $\mathbf{X}$. Sean $p_G$ y $p_{G'}$ las distribuciones de probabilidad que se factorizan según los grafos $G$ y $G'$, respectivamente, y se componen como $p_G(X|pa_G(X)) = p(X|pa_G(X))$ y $p_{G'}(X|pa_{G'}(X)) = p(X|pa_{G'}(X))$, $\forall X \in \mathbf{X}$. Si $p_G(C|\mathbf{u}) = p_{G'}(C|\mathbf{u})$ $\forall \mathbf{u}$, se dice que $G$ y $G'$ son *equivalentes en clasificación.*

**Proposición 6.1.1** *Dado cualquier DAG $G = (\mathbf{X}, E_G)$, sea $G_c = (\mathbf{X}, E_c)$ el subgrafo de $G$ que se define como sigue:*

1. *$Pa_{G_c}(C) = Pa_G(C)$, es decir, $C$ tiene el mismo conjunto de padres en ambos grafos.*

2. *$\forall X \in \mathbf{X}$, $X \neq C$, si $C \in Pa_G(X)$ entonces $Pa_{G_c}(X) = Pa_G(X)$, esto es, en caso de que $C$ sea padre de $X$, entonces $X$ tiene el mismo conjunto de padres en ambos grafos.*

3. *$\forall X \in \mathbf{X}$, $X \neq C$, si $C \notin Pa_G(X)$ entonces $Pa_{G_c}(X) = \emptyset$, esto quiere decir, que en caso de que $C$ no sea padre de $X$, entonces $X$ no tiene padres en $G_C$.*

*En ese caso $G$ y $G_c$ son equivalentes en clasificación. Es más, para cualquier otro subgrafo $H = (\mathbf{X}, E_H)$ de $G$ tal que $G$ y $H$ sean equivalentes en clasificación, entonces $G_c$ es también un subgrafo de $H$.*

**Demostración.** La probabilidad condicional $p_G(C|x_1, \ldots, x_n)$ se puede expresar en términos de las probabilidades $p(C|pa_G(C))$ y $p(X|pa_G(X))$, $\forall X$

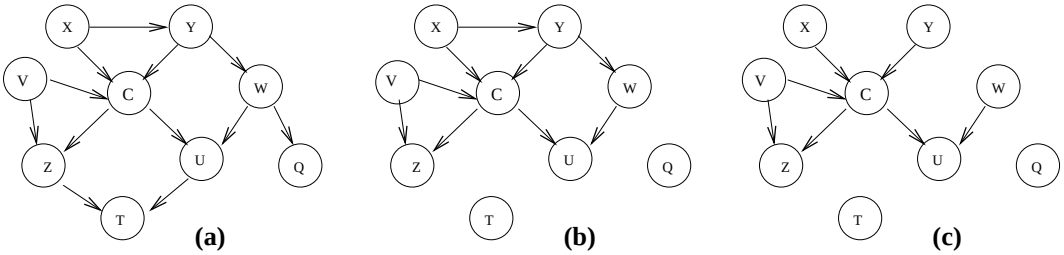


Figura 6.1: (a) Un DAG $G$; (b) subgrafo de $G$ inducido a partir del manto de Markov de la variable $C$; (c) subgrafo $G_c$.

tal que $C \in Pa_G(X)$. Como el conjunto de padres $Pa_G(C)$ y $Pa_G(X)$ son iguales en el subgrafo $G_c$, la primera afirmación se verifica inmediatamente.

Además, estos conjuntos de padres también son los mismos para cualquier subgrafo de $G$ equivalente en clasificación. Como $Pa_{G_c}(Y) = \emptyset$ para cualquier otro nodo $Y$, se verifica que $G_c$ es un subgrafo de $H$. ∎

Cabe señalar que $G_c$ no es igual al subgrafo de $G$ inducido por $C$ y el manto de Markov de $C$, ya que el primero contiene un número menor de arcos que el segundo. La figura 6.1 ilustra las diferencias entre estos dos subgrafos.

La proposición 6.1.1 identifica el mínimo subgrafo de cualquier grafo dirigido acíclico $G$ que actúa exactamente como $G$ a la hora de clasificar la variable $C$. Esto nos sugiere que podríamos tratar de encontrar un buen clasificador si la búsqueda se centra en el conjunto de DAGs de este tipo, en lugar de buscar en todo el espacio de DAGs. Pero para ello, en primer lugar, vamos a definir formalmente este tipo de estructura.

**Grafo dirigido acíclico centrado en la clase (C-DAG)** Un DAG $G = (\mathbf{X}, E_G)$ es un grafo dirigido acíclico *centrado en la clase* (abreviadamente, C-DAG), con respecto a la variable $C$, si y sólo si satisface la siguiente condición:

$$\forall X, Y \in \mathbf{X},\ \text{si } X \rightarrow Y \in E_G \text{ entonces } Y = C \text{ o } X = C \text{ o } C \rightarrow Y \in E_G\,.$$

El C-DAG puede contener sólo los arcos que enlacen la variable clase con cualquier atributo (en cualquier dirección) y arcos entre los atributos sólo en el caso de que la clase sea también padre del atributo hijo.

Un DAG centrado en la clase puede verse como la representación canónica de una clase de grafos dirigidos acíclicos, todos ellos equivalentes como clasificadores de $C$. El siguiente resultado formaliza esta idea:

**Proposición 6.1.2** *Dado cualquier C-DAG $H = (\mathbf{X}, E_H)$, sea $\mathcal{C}_H$ el conjunto*

$$\mathcal{C}_H = \{G \mid G \text{ es un DAG sobre } \mathbf{X} \text{ y } G_c = H\}.$$

*En ese caso*

1. *$G \in \mathcal{C}_H$ si y sólo si $Pa_G(X) = Pa_H(X)\ \forall X \in \mathbf{X}$ tal que $C \in Pa_G(X)$, y $Pa_G(C) = Pa_H(C)$.*

2. *La familia de conjuntos $\{\mathcal{C}_H \mid H$ es un C-DAG sobre $\mathbf{X}\}$ es una partición del conjunto de DAGs sobre $\mathbf{X}$.*

**Demostración.** La primera afirmación es obvia.

Para la segunda, se ve claramente que para cualquier DAG $G$, podemos encontrar un C-DAG $H$ tal que $G \in \mathcal{C}_H$: $H = G_c$. Además, si $G \in \mathcal{C}_H \cap \mathcal{C}_{H'}$, entonces $H = G_c = H'$, y $\{\mathcal{C}_H\}$ es una partición. ∎

### 6.1.2. Grafos parcialmente dirigidos, acíclicos y restringidos (RPDAGs).

Aunque el uso de C-DAGs en lugar de los DAGs habituales va a reducir el espacio de búsqueda, se puede conseguir una mayor reducción si tenemos en cuenta además el concepto de clases de DAGs *equivalentes en independencia* [346], esto es, DAGs donde cada uno representa un conjunto diferente de restricciones de independencia condicional. En el caso de que todas las variables sean discretas, como es nuestro caso, la *equivalencia en independencia* coincide con el concepto de *equivalencia en distribución*, en el sentido de que cada clase representa un conjunto diferente de distribuciones de probabilidad.

Los objetos gráficos utilizados para representar clases de DAGs equivalentes en independencia son grafos acíclicos parcialmente dirigidos, (abreviadamente *PDAGs*, del inglés *partially directed acyclic graphs*). Estos grafos pueden contener tanto enlaces dirigidos (arcos), como no dirigidos (enlaces), pero no pueden tener ciclos dirigidos.

Hay un tipo de PDAGs que pueden ser usados para representar canónicamente clases de DAGs equivalentes en independencia: son los denominados *PDAGs completos* [73] conocidos también como *grafos esenciales* [13, 83] o *patrones* [323]. Sin embargo, los PDAGs completos son considerablemente más complicados que los PDAGs comunes (una caracterización de los PDAGs completos se puede encontrar en [13]). En el contexto del aprendizaje de redes bayesianas, se han desarrollado una serie de algoritmos que llevan a cabo la búsqueda en un espacio de clases de DAGs equivalentes en independencia [13, 73, 83, 224, 323]. Este criterio reduce el tamaño del espacio de búsqueda[1], el cual ahora tiene una topología más suave, y evita tomar decisiones prematuras sobre la dirección de los primeros arcos.

El precio que deben pagar estos algoritmos por esta reducción del espacio de búsqueda, es que la evaluación de estructuras candidatas no se benefician de la propiedad de la descomposición de muchas de las métricas (por lo que, el funcionamiento de estos algoritmos es menos eficiente), no obstante este inconveniente ya se ha superado [6, 74], pues también se han desarrollado para espacios de búsqueda en clases de equivalencia métodos que para pasar de una configuración a otra vecina no necesitan efectuar todas las evaluaciones locales, sino sólo un número muy limitado.

Nuestra propuesta es aplicar estas ideas acerca de grafos acíclicos dirigidos equivalentes en independencias a los DAGs centrados en la clase. Aunque pueden utilizarse PDAGs completos, nosotros utilizaremos un esquema de representación no canónica de los *PDAGs restringidos* (abreviadamente, *RPDAGs*, del inglés Restricted PDAG) [6] que son considerablemente más simples que los PDAGs completos.

Los RPDAGs son ligeramente diferentes del formalismo de los PDAGs completos: dos RPDAGs diferentes pueden corresponder a la misma clase de DAGs equivalentes en independencias, es decir, la correspondencia entre RPDAGs y clases de equivalencia en DAGs no es uno a uno. Permitiendo que clases de DAGs equivalentes sean representados (en algunos casos) por RPDAGs distintos, ganamos en eficiencia al explorar el espacio de búsqueda. Un enlace aislado $X$—$Y$ junto con otro enlace $X$—$Z$ representa cualquier combinación de arcos excepto aquellos que crean nuevos patrones cabeza-cabeza (es decir, mientras $Y$ y $Z$ no estén unidos, serían $Y \rightarrow X \rightarrow Z$, $Y \leftarrow X \rightarrow Z$ y $Y \leftarrow X \leftarrow Z$).

[1] Aunque esta reducción no es tan importante como se esperaba [135].

Vamos a introducir algunas notaciones adicionales y, a continuación, el concepto de RPDAG.

El *esqueleto* de un DAG es el grafo no dirigido que resulta de ignorar la direccionalidad de cada arco. Como vimos, un *patrón cabeza-cabeza* en un DAG $G$ es un triplete ordenado de nodos, $(X, Y, Z)$, de manera que $G$ contiene los arcos $X \rightarrow Y$ y $Y \leftarrow Z$, y además $X$ y $Z$ no están unidos. Dado un PDAG $G = (\mathbf{X}, E_G)$, para cada nodo $X \in \mathbf{X}$ se definen los siguientes subconjuntos:

- $Pa_G(X) = \{Y \in \mathbf{X} \mid Y \rightarrow X \in E_G\}$, el conjunto de los *padres* de $X$.
- $Ch_G(X) = \{Y \in \mathbf{X} \mid X \rightarrow Y \in E_G\}$, el conjunto de *hijos* de $X$.
- $Sib_G(X) = \{Y \in \mathbf{X} \mid X — Y \in E_G\}$, el conjunto de *hermanos* de $X$.
- $Ad_G(X) = Pa_G(X) \cup Ch_G(X) \cup Sib_G(X)$, el conjunto de *adyacentes* de $X$.

**PDAG restringido [6]** Un PDAG $G = (\mathbf{X}, E_G)$ es un PDAG restringido (abreviadamente RPDAG, del inglés *Restricted PDAG*) si y sólo si satisface las siguientes condiciones:

1. $\forall X \in \mathbf{X}$, si $Pa_G(X) \neq \emptyset$ entonces $Sib_G(X) = \emptyset$.
2. $G$ no contiene ningún ciclo dirigido, es decir, un ciclo que contenga tan solo enlaces dirigidos.
3. $G$ no contiene ningún ciclo completamente no dirigido, esto es, un ciclo que contenga tan solo enlaces no dirigidos.
4. Si $X \rightarrow Y \in E_G$ entonces $|Pa_G(Y)| \geq 2$, o bien, $Pa_G(X) \neq \emptyset$. ∎

La primera condición en la definición de RPDAG nos dice que la configuración $Y \rightarrow X — Z$ no ocurre en un RPDAG $G$. La cuarta condición establece que un arco $X \rightarrow Y$ existe en $G$ sólo si es parte de un patrón cabeza-cabeza o hay otro arco (originado por un patrón cabeza-cabeza) apuntando a $X$.

Como los RPDAGs son representaciones de conjuntos de DAGs equivalentes en independencia, debemos definir qué conjunto de DAGs es representado

por un determinado RPDAG $G$, esto es, qué dirección puede aplicarse a los enlaces de $G$ para obtener un DAG (las condiciones 2 y 3 de la definición de RPDAG garantizan que esto es siempre posible). La forma de obtener un DAG a partir de un RPDAG puede ser definida como:

**Extensión de un RPDAG [6]** Dado un RPDAG $G = (\mathbf{X}, E_G)$, decimos que un DAG $H = (\mathbf{X}, E_H)$ es una extensión de $G$ si y sólo si:

1. $G$ y $H$ tienen el mismo esqueleto.
2. En el caso de que $X \rightarrow Y \in E_G$ entonces $X \rightarrow Y \in E_H$ (ningún arco es redirigido).
3. $G$ y $H$ tienen los mismos patrones cabeza-cabeza. ∎

Vamos a notar $Ext(G)$ al conjunto de DAGs que son extensiones de un determinado RPDAG $G$.

Se puede demostrar [6] que (1) $Ext(G) \neq \emptyset$, (2) $\forall H, H' \in Ext(G)$ $H$ y $H'$ son equivalentes en independencia, y (3) $H, H' \in Ext(G)$ si y sólo si $H$ y $H'$ tienen el mismo esqueleto y los mismos patrones cabeza-cabeza.

### 6.1.3. RPDAGs centrados en la clase (C-RPDAGs).

Volviendo a nuestro problema de búsqueda de clasificadores basados en redes bayesianas, nuestra propuesta consiste en definir el espacio de búsqueda como el conjunto de RPDAGs que son diferentes desde el punto de vista de la clasificación. Más formalmente, vamos a definir:

**RPDAG centrado en la clase** Un RPDAG centrado en la clase (abreviadamente, C-RPDAG), es un RPDAG cuyas extensiones son DAGs centrados en la clase.

La siguiente proposición nos permite caracterizar el concepto de C-RPDAG.

**Proposición 6.1.3** *Un PDAG $G$ es un RPDAG centrado en la clase si y sólo si se cumple las siguientes condiciones:*

1. *$G$ no contiene ningún ciclo dirigido.*
2. *Si $Pa_G(C) \neq \emptyset$ entonces $|Pa_G(C)| \geq 2$ y $Sib_G(C) = \emptyset$.*

3. $\forall X \in \mathbf{X}$, $X \neq C$, *si* $Pa_G(X) \neq \emptyset$ *entonces* $C \in Pa_G(X)$ *y o bien* $|Pa_G(X)| \geq 2$ *o bien* $|Pa_G(C)| \geq 2$.

4. $\forall X \in \mathbf{X}$, $X \neq C$, *si* $Sib_G(X) \neq \emptyset$ *entonces* $Sib_G(X) = \{C\}$ *y* $Pa_G(X) = \emptyset$.

**Demostración.** *Condición necesaria*: se asume que $G$ es un C-RPDAG y se probarán las cuatro condiciones 1–4 de la proposición 6.1.3.
*Condición 1*: esto es obvio a partir de la condición 2 de la definición de RPDAG
*Condición 2*: si $Pa_G(C) \neq \emptyset$, entonces $Sib_G(C) = \emptyset$ usando la primera condición de la definición de RPDAG. Además, hay al menos un arco $X \rightarrow C$ en $G$. Usando la condición 4 de la definición de RPDAG, se obtiene $|Pa_G(C)| \geq 2$ o $Pa_G(X) \neq \emptyset$. Si $Pa_G(X) \neq \emptyset$, entonces se tiene un arco $Y \rightarrow X$ tal que $Y \neq C$, $X \neq C$ y el arco $C \rightarrow X$ no existe, lo cual contradice el hecho de que las extensiones de $G$ son C-DAGs. Por tanto, $|Pa_G(C)| \geq 2$.
*Condición 3*: si $Pa_G(X) \neq \emptyset$, $X \neq C$, entonces hay un arco $Y \rightarrow X$ en $G$ y en cualquier extensión de $G$. Como estas extensiones son C-DAGs, entonces o bien $Y = C$ o $\exists C \rightarrow X$. En todo caso $C \in Pa_G(X)$. Usando la condición 4 de la definición de RPDAG $|Pa_G(X)| \geq 2$ o $Pa_G(C) \neq \emptyset$. En este último caso, la condición 2 asegura que $|Pa_G(C)| \geq 2$.
*Condición 4*: si $Sib_G(X) \neq \emptyset$, $X \neq C$, entonces se obtiene $Pa_G(X) = \emptyset$, usando la condición 1 de la definición de RPDAG. Además, un enlace $Y$—$X$ existe en $G$. Este enlace será un arco en las extensiones de $G$. Como estas extensiones son C-DAGs, la única opción posible es que o bien $Y = C$ o el patrón cabeza-cabeza $C \rightarrow Y \leftarrow X$ existe en alguna extensión de $G$. Esta última opción podría implicar que el patrón cabeza-cabeza también existe en $G$, y esto contradice el hecho de que $Y$—$X$ es un enlace y no un arco en $G$. Por lo tanto, $C \in Sib_G(X)$ y $C$ es el único hermano de $X$.

*Condición suficiente*: se asumen las cuatro condiciones en la proposición 6.1.3, y se van a probar las cuatro condiciones en la definición de RPDAG y que las extensiones de $G$ son C-DAGs.
*Condición 1*: se asume que $Pa_G(X) \neq \emptyset$. Si $X = C$ la condición 2 de la proposición 6.1.3 garantiza que $Sib_G(X) = \emptyset$; si $X \neq C$, entonces la condición 3 de la proposición 6.1.3 afirma que $C \in Pa_G(X)$, por lo que $C \notin Sib_G(X)$. De la condición 4 de la proposición 6.1.3 se puede obtener ahora que $Sib_G(X) = \emptyset$.
*Condición 2*: se verifica inmediatamente a partir de la condición 1 de la proposición 6.1.3.

*Condición 3*: si existe un ciclo completamente no dirigido en $G$, entonces al menos tres nodos tienen más de un nodo hermano. No obstante, a partir de la condición 4 de la proposición 6.1.3 sabemos que todos los nodos, excepto la clase, tienen como mucho un hermano.
*Condición 4*: se asume que el arco $X \rightarrow Y$ existe en $G$. Si $Y = C$ entonces $Pa_G(C) \neq \emptyset$, y la condición 2 de la proposición 6.1.3 asegura que $|Pa_G(Y)| = |Pa_G(C)| \geq 2$. Si $X = C$ entonces $Pa_G(Y) \neq \emptyset$ y la condición 3 de la proposición 6.1.3 asegura que $|Pa_G(Y)| \geq 2$ o $|Pa_G(X)| = |Pa_G(C)| \geq 2$. Si $X \neq C$ y $Y \neq C$ entonces de la condición 3 de la proposición 6.1.3 se sabe que $C \in Pa_G(Y)$, por lo que $|Pa_G(Y)| \geq 2$.

Por tanto, $G$ es un RPDAG. Para probar que también es un C-RPDAG, veremos que sus extensiones son C-DAGs. Sea $H$ una extensión de $G$ y supongamos que $X \rightarrow Y$ es un arco en $H$ y $X \neq C$. Este arco puede corresponderse con el arco $X \rightarrow Y$ o el enlace $X$—$Y$ en $G$. En el primer caso $Pa_G(Y) \neq \emptyset$, y de la condición 3 de la proposición 6.1.3 se obtiene que $C \in Pa_G(Y)$, por lo que el arco $C \rightarrow Y$ existe en $G$. En el segundo caso, como $Sib_G(X) \neq \emptyset$, a partir de la condición 4 de la proposición 6.1.3 se obtiene que $Sib_G(X) = \{C\}$, por lo que $Y = C$. Por tanto, $G$ es un C-DAG. ∎

En la figura 6.2 se muestran algunos ejemplos de C-RPDAGs. Otro resultado interesante acerca de los C-RPDAGs puede extraerse de la proposición siguiente:

**Proposición 6.1.4** *Sea $G = (\mathbf{X}, E_G)$ un C-RPDAG. Entonces, $\forall H, H' \in Ext(G)$, $H$ y $H'$ son equivalentes en clasificación.*

**Demostración.** Como $H$ y $H'$ son extensiones de un RPDAG, son equivalentes en independencias, por lo que $p_H(c, x_1, \ldots, x_n) = p_{H'}(c, x_1, \ldots, x_n)$, $\forall c, x_1, \ldots, x_n$, y por tanto $p_H(c|x_1, \ldots, x_n) = p_{H'}(c|x_1, \ldots, x_n)$. ∎

De esta manera, los elementos en nuestro espacio de búsqueda de C-RPDAGs representan conjuntos de C-DAGs equivalentes en clasificación, cada uno a su vez representa un conjunto de DAGs equivalentes en clasificación. Cabe señalar que, al igual que pasaba con los RPDAGs, los C-RPDAGs no son siempre representaciones canónicas de la clase de DAGs equivalentes en clasificación, es decir, dos DAGs equivalentes en clasificación pueden estar asociados a diferentes C-RPDAGs. En cualquier caso, se obtiene una reducción notable de configuraciones en el espacio de búsqueda respecto a los

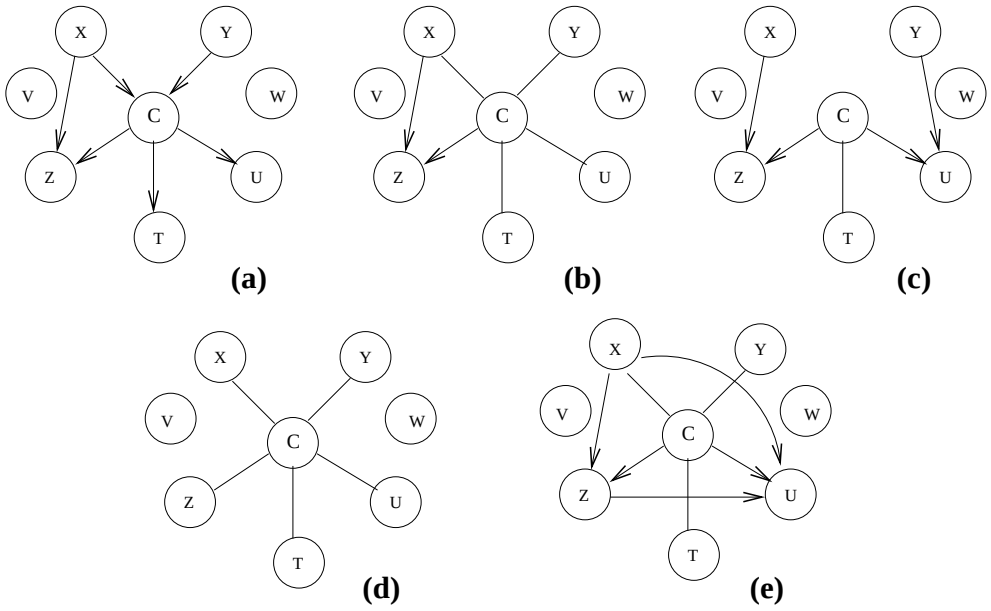


Figura 6.2: Ejemplos de C-RPDAGs.

DAGS.

Creemos que ésta no es la única ventaja de la utilización de los C-RPDAGs. Por un lado, un proceso de búsqueda en el espacio de DAGs puede desorientarse con decisiones que no influyen en el clasificador final obtenido, en el sentido de que la inclusión de algunos arcos, irrelevantes para la tarea de clasificación, puede impedir o dificultar la inclusión de otros arcos que sí sean relevantes. No esperamos que este problema se produzca en un proceso de búsqueda orientado hacia la clasificación, tal como el basado en C-RPDAGs.

Por otro lado, respecto a los métodos de búsqueda basados en DAGs aumentados, aquellos clasificadores en los que $C \in Pa_G(X)\ \forall X \in \mathbf{X} \setminus \{C\}$ (como TAN, FAN y BAN), nuestro espacio de búsqueda no impone la subestructura del naïve bayes, aunque éste pueda obtenerse si fuera la estructura realmente apropiada (obsérvese que un DAG aumentado es siempre un C-DAG). También podemos obtener estructuras más generales, como aquellas en las que un nodo es padre de un hijo de la clase, y este nodo no está conectado a la clase.

Cabe señalar que la selección de características es también una parte integral de este enfoque, ya que todas las variables que queden aisladas al final del proceso de búsqueda serán ignoradas por el clasificador. Por otra parte, las estructuras obtenidas por otros métodos de clasificación pueden obtenerse también mediante C-RPDAGs. Este es el caso, por ejemplo, del clasificador semi naïve bayes y el clasificador naïve bayes selectivo en sus diferentes formalismos.

## 6.2. El método de búsqueda.

Vamos a utilizar un método local para explorar el espacio de búsqueda de los C-RPDAGs. El punto de partida del proceso de búsqueda será un C-RPDAG vacío (que corresponde a un DAG vacío). Tendremos que definir los operadores para pasar de una configuración a otra configuración vecina.

### 6.2.1. Los operadores.

Nuestros operadores básicos son esencialmente los mismos que los utilizados [6] para RPDAGs, esto es, la inclusión de un enlace entre un par de nodos no adyacentes y la eliminación de un enlace entre un par de nodos vecinos. Estos enlaces pueden ser dirigidos o no dirigidos.

En el caso de añadir un enlace, podemos obtener diferentes configuraciones vecinas, dependiendo de la topología del C-RPDAG actual y de la dirección del arco incluido: si estamos probando la inclusión de un enlace $X$—$Y$ entre dos nodos $X$ e $Y$, no adyacentes, puede provocar que se prueben la validez de diferentes configuraciones obtenidas por la inserción del enlace: el arco $X \rightarrow Y$, el arco $X \leftarrow Y$, el patrón cabeza-cabeza $X \rightarrow Y \leftarrow Z$ o el patrón cabeza-cabeza $Z \rightarrow X \leftarrow Y$, donde $Z$ en los dos últimos casos, podría ser cualquier nodo que cumpla que el enlace $Y$—$Z$, o el enlace $Z$—$X$ existe en la configuración actual. Sin embargo, la eliminación de un enlace siempre resultará en sólo una configuración vecina.

Los cinco operadores utilizados por Acid y de Campos en [6] para moverse en el espacio de los RPDAGs eran: A_arc$(X, Y)$, añadir un arco $X \rightarrow Y$; A_link$(X, Y)$, añadir un enlace $X$—$Y$; D_arc$(X, Y)$, borrar un arco $X$—$Y$; D_link$(X, Y)$, borrar un enlace $X$—$Y$; A_hh$(X, Y, Z)$, añadir un arco $X \rightarrow Y$

y crear un patrón cabeza-cabeza $X \rightarrow Y \leftarrow Z$ transformando el enlace $Y$—$Z$ en el arco $Y \leftarrow Z$.

Sin embargo, teniendo en cuenta el carácter especial de la variable de clase $C$ en nuestro caso, estos operadores se han reformulado en términos de los padres de $C$, los hijos de $C$ y los padres de los hijos de $C$, con el fin de dejar claro su significado desde la perspectiva de la variable clase. Los expondremos detalladamente:

- A_ParentOfC($X$), añadir el arco $X \rightarrow C$.
- A_ChildOfC($X$), añadir el arco $C \rightarrow X$.
- A_SiblingOfC($X$), añadir el enlace $X$—$C$.
- A_HHOfC($X, Y$), crear un patrón cabeza-cabeza $X \rightarrow C \leftarrow Y$, al añadir un arco $X \rightarrow C$ y transformar el enlace $Y$—$C$ en el arco $Y \rightarrow C$.
- A_ParentOfChild($X, Y$), añadir el arco $X \rightarrow Y$.
- A_HHOfChild($X, Y$), crear un patrón cabeza-cabeza $X \rightarrow Y \leftarrow C$, al añadir un arco $X \rightarrow Y$ y transformar el enlace $C$—$Y$ en el arco $C \rightarrow Y$.
- D_ParentOfC($X$), borrar el arco $X \rightarrow C$.
- D_ChildOfC($X$), borrar el arco $C \rightarrow X$, junto con todos los arcos que apuntan a $X$.
- D_SiblingOfC($X$), borrar el enlace $X$—$C$.
- D_HHOfC($X, Y$), eliminación del patrón cabeza-cabeza $X \rightarrow C \leftarrow Y$, al borrar el arco $X \rightarrow C$ y transformar el arco $Y \rightarrow C$ en el enlace $Y$—$C$.
- D_ParentOfChild($X, Y$), borrar el arco $X \rightarrow Y$.
- D_HHOfChild($X, Y$), eliminación del patrón cabeza-cabeza $X \rightarrow Y \leftarrow C$, al borrar el arco $X \rightarrow Y$ y transformar el arco $C \rightarrow Y$ en el enlace $Y$—$C$.

Las condiciones que el actual C-RPDAG $G$ debe verificar para que cada uno de estos operadores se pueden aplicar con el fin de obtener un C-RPDAG $G'$, vecino válido, se muestran en la tabla 6.1. Estas condiciones pueden deducirse fácilmente de la proposición 6.1.3.

La tabla 6.1 también muestra las acciones necesarias para transformar $G$ en el correspondiente vecino $G'$, incluidas las acciones necesarias para garantizar que $G'$ sea un C-RPDAG. En las figuras 6.3 y 6.4 se muestra un ejemplo para cada uno de los operadores.

Notemos que, como cada uno de los operadores de inserción tiene un operador correspondiente para su borrado, podemos obtener cualquier C-RPDAG a partir de cualquier otro C-RPDAG.

Tabla 6.1: Los operadores, sus condiciones de aplicabilidad y las acciones necesarias.

| Operador | Condiciones | Acciones |
|---|---|---|
| A_ParentOfC($X$) | $X \notin Ad_G(C)$; $Pa_G(C) \neq \emptyset$ | insertar($X \rightarrow C$) |
| A_ChildOfC($X$) | $X \notin Ad_G(C)$; $Pa_G(C) \neq \emptyset$ | insertar($C \rightarrow X$) |
| A_SiblingOfC($X$) | $X \notin Ad_G(C)$; $Pa_G(C) = \emptyset$ | insertar($X$—$C$) |
| A_HHOfC($X, Y$) | $X \notin Ad_G(C)$; $Pa_G(C) = \emptyset$<br>$Y \in Sib_G(C)$ | insertar($X \rightarrow C$)<br>borrar($Y$—$C$); insertar($Y \rightarrow C$)<br>$\forall Z \in Sib_G(C) \setminus \{Y\}$ {<br>borrar($C$—$Z$); insertar($C \rightarrow Z$)} |
| A_ParentOfChild($X, Y$) | $X \notin Ad_G(Y)$; $Y \in Ch_G(C)$<br>No hay un camino dirigido<br>de $Y$ a $X$ en $G$ | insertar($X \rightarrow Y$) |
| A_HHOfChild($X, Y$) | $X \notin Ad_G(Y)$; $Y \in Sib_G(C)$<br>No hay un camino dirigido<br>de $Y$ a $X$ en $G$ | insertar($X \rightarrow Y$)<br>borrar($C$—$Y$)<br>insertar($C \rightarrow Y$) |
| D_ParentOfC($X$) | $X \in Pa_G(C)$; $\lvert Pa_G(C)\rvert \geq 3$ | borrar($X \rightarrow C$) |
| D_ChildOfC($X$) | $X \in Ch_G(C)$ | $\forall Z \in Pa_G(X)$ borrar($Z \rightarrow X$) |
| D_SiblingOfC($X$) | $X \in Sib_G(C)$ | borrar($X$—$C$) |
| D_HHOfC($X, Y$) | $X \in Pa_G(C)$; $Y \in Pa_G(C)$<br>$\lvert Pa_G(C)\rvert = 2$ | borrar($X \rightarrow C$)<br>borrar($Y \rightarrow C$); insertar($Y$—$C$)<br>$\forall Z \in Ch_G(C)$<br>si $Pa_G(Z) = \{C\}$ {<br>borrar($Z \rightarrow C$); insertar($Z$—$C$)} |
| D_ParentOfChild($X, Y$) | $X \in Pa_G(Y)$; $Y \in Ch_G(C)$<br>$\lvert Pa_G(Y)\rvert \geq 3$ o $Pa_G(C) \neq \emptyset$ | borrar($X \rightarrow Y$) |
| D_HHOfChild($X, Y$) | $X \in Pa_G(Y)$; $Y \in Ch_G(C)$<br>$\lvert Pa_G(Y)\rvert < 3$; $Pa_G(C) = \emptyset$ | borrar($X \rightarrow Y$)<br>borrar($C \rightarrow Y$); insertar($C$—$Y$) |

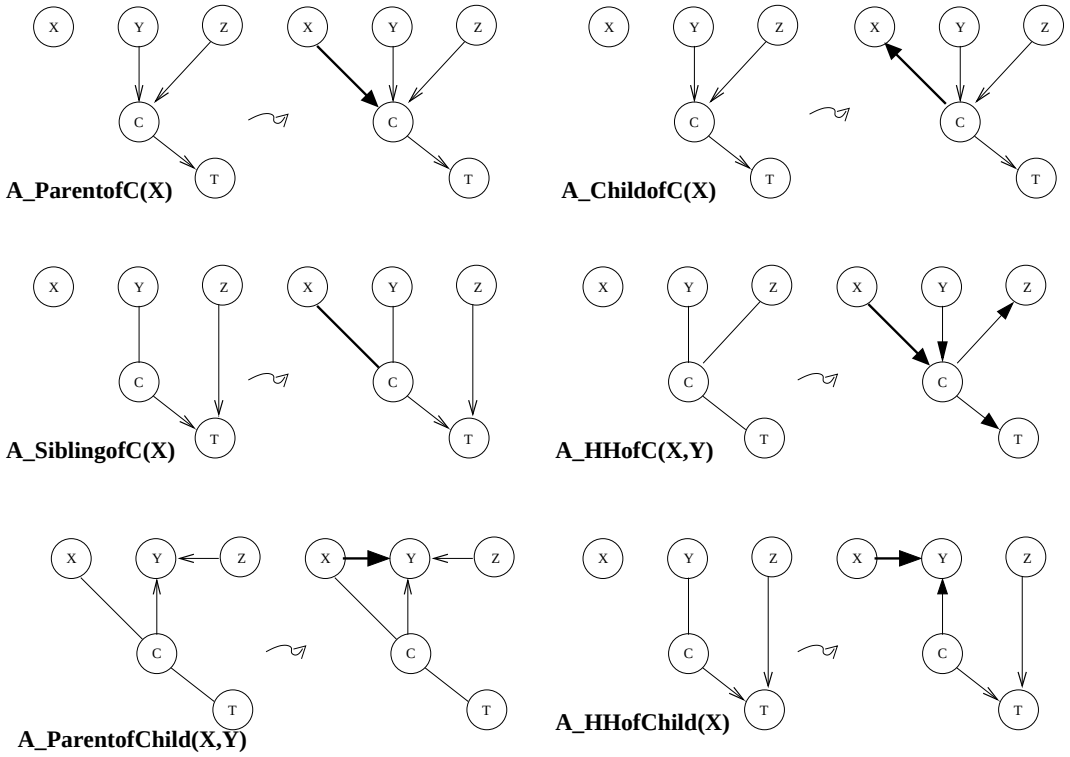


Figura 6.3: Ejemplos de los seis operadores de inserción.

### 6.2.2. Evaluación de las estructuras candidatas.

Para obtener el clasificador, el método de búsqueda que hemos descrito puede aplicarse en combinación con cualquier función capaz de medir el ajuste entre cada C-RPDAG explorado y los datos disponibles. Para esto se pueden considerar varias opciones.

Una primera opción es utilizar la precisión del clasificador resultante para la evaluación de cada red candidata. Este es el enfoque *de envoltura*, que comúnmente se utiliza en muchos algoritmos para la obtención de clasificadores o para la selección de características [176, 182, 206, 205, 260, 315]. Como este tipo de medida tiende a causar problemas de sobreajuste, la solución habitual es utilizar un esquema de estimación, como validación cruzada, para evaluar la exactitud de predicción del clasificador candidato sobre diferentes subconjuntos de datos. Este enfoque es muy costoso, a menos que el modelo

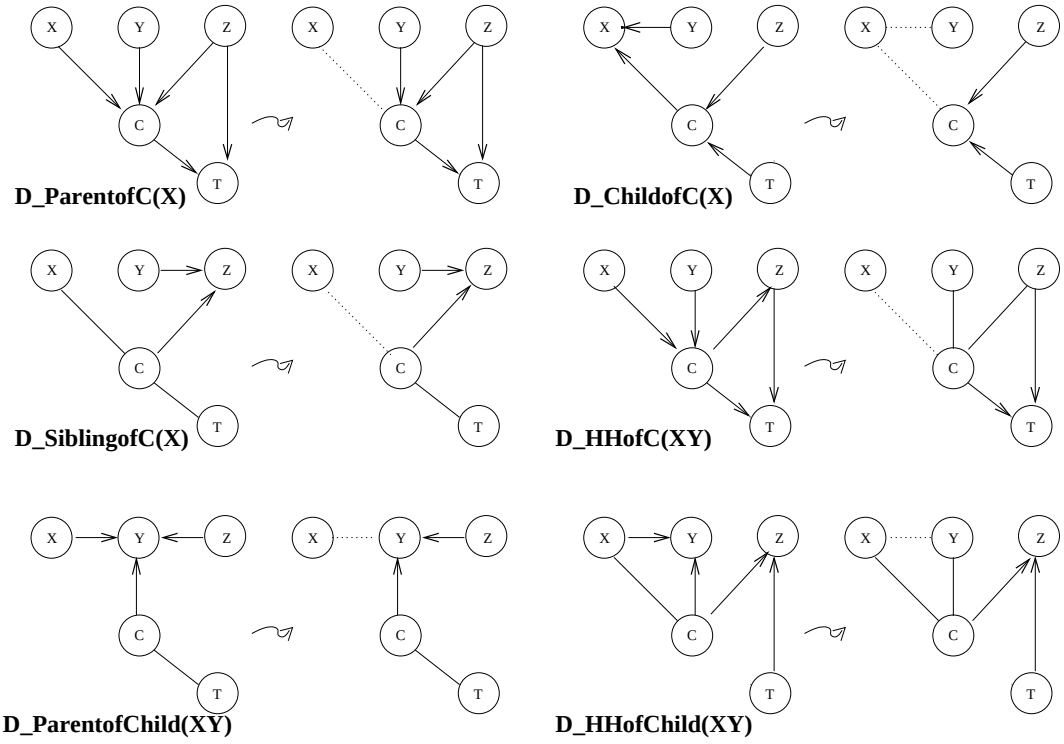


Figura 6.4: Ejemplos de los seis operadores de borrado.

seleccionado sea simple, como el clasificador naïve bayes.

Una segunda opción consistiría en desarrollar una métrica especializada que pudiera considerar lo bien que una red describe la distribución de probabilidad de la variable clase dados los atributos. Lamentablemente, los intentos en esta dirección hasta la fecha han resultado computacionalmente no operativos [118].

Aunque no se descarta la posibilidad de proseguir en este enfoque en el futuro, se ha considerado un opción más sencilla y eficiente: no especializar la función de ajuste, tal como sucede con los algoritmos de aprendizaje de redes bayesianas sin restricciones. Ejemplos de estas funciones son las conocidas métricas BIC, BDe [150], K2 [80] o MDL [204].

Se ha argumentado [118] que estas métricas miden el ajuste entre la distribución conjunta asociada a la red y los datos, pero no miden el ajuste entre la distribución condicional de la clase dado los atributos y los datos.

Por consiguiente, su uso puede dar lugar a clasificadores pobres. Si bien la primera de estas afirmaciones es, sin duda, cierta, la segunda es discutible. Los resultados experimentales de la sección 6.3 indican que algunas funciones de ajuste no especializadas, tales como BDe, en combinación con el espacio de búsqueda de C-RPDAGs, son adecuadas para tareas de clasificación.

En principio, una función de ajuste o métrica, $g$, evalúa DAGs. Sin embargo, los elementos en nuestro espacio de búsqueda no son DAGs sino clases de DAGs equivalentes (más precisamente C-RPDAGs). Una manera sencilla de utilizar nuestro método sería seleccionar una extensión $H$ de un C-RPDAG candidato $G$ a ser evaluado, y calcular $g(H : D)$. Para ello, sería razonable utilizar una función de ajuste *equivalente en valor* [2], para garantizar que se obtiene el mismo resultado independientemente del DAG $H$ seleccionado.

Por otra parte, en aras de la eficiencia en el proceso de evaluación, la métrica debería de ser *descomponible*. Una función $g$ es descomponible cuando el valor de la función para cualquier estructura de red bayesiana puede expresarse como combinación (producto o suma en el espacio logarítmico) de los valores locales para cada familia de nodos, un nodo y sus padres, es decir,

$$g(G : D) = \sum_{X \in \mathbf{X}} g_D(X, Pa_G(X))$$

De esta manera el DAG obtenido por la inserción o el borrado de un arco sobre el DAG actual $H$ puede ser evaluado simplemente mediante la modificación de un solo valor local.

Para los operadores que se están considerando en nuestro espacio de C-RPDAG, se puede obtener esa misma propiedad: mediante el uso de una función de ajuste equivalente en valor y descomponible, en primer lugar, no es necesario transformar un C-RPDAG en un DAG, sino que el C-RPDAG puede ser evaluado directamente, y en segundo lugar, el valor de ajuste de cualquier C-RPDAG vecino puede obtenerse mediante el cálculo de dos evaluaciones locales como mucho. De esta manera, se conservan todas las ventajas de los métodos de búsqueda en el espacio de DAGs, pero en un espacio de búsqueda más reducido y más robusto. La siguiente proposición demuestra estas afirmaciones.

[2] una función que dé la misma puntuación a DAGs equivalentes en independencia.

**Proposición 6.2.1** *Sea $G$ un C-RPDAG y $G'$ cualquier C-RPDAG que se obtiene mediante la aplicación de alguno de los operadores descritos en la Tabla 6.1 sobre $G$. Sea $g$ una función de ajuste equivalente en valor y descomponible .*

*(a) Si el operador es A_ParentOfC$(X)$ entonces*
$g(G' : D) = g(G : D) - g_D(C, Pa_G(C)) + g_D(C, Pa_G(C) \cup \{X\})$

*(b) Si el operador es A_ChildOfC$(X)$ entonces*
$g(G' : D) = g(G : D) - g_D(X, \emptyset) + g_D(X, \{C\})$

*(c) Si el operador es A_SiblingOfC$(X)$ entonces*
$g(G' : D) = g(G : D) - g_D(C, \emptyset) + g_D(C, \{X\})$

*(d) Si el operador es A_HHOfC$(X, Y)$ entonces*
$g(G' : D) = g(G : D) - g_D(C, \{Y\}) + g_D(C, \{X, Y\})$

*(e) Si el operador es A_ParentOfChild$(X, Y)$ entonces*
$g(G' : D) = g(G : D) - g_D(Y, Pa_G(Y)) + g_D(Y, Pa_G(Y) \cup \{X\})$

*(f) Si el operador es A_HHOfChild$(X, Y)$ entonces*
$g(G' : D) = g(G : D) - g_D(Y, \{C\}) + g_D(Y, \{C, X\})$

*(g) Si el operador es D_ParentOfC$(X)$ entonces*
$g(G' : D) = g(G : D) - g_D(C, Pa_G(C)) + g_D(C, Pa_G(C) \setminus \{X\})$

*(h) Si el operador es D_ChildOfC$(X)$ entonces*
$g(G' : D) = g(G : D) - g_D(X, Pa_G(X)) + g_D(X, \emptyset)$

*(i) Si el operador es D_SiblingOfC$(X)$ entonces*
$g(G' : D) = g(G : D) - g_D(C, \{X\}) + g_D(C, \emptyset)$

*(j) Si el operador es D_HHOfC$(X, Y)$ entonces*
$g(G' : D) = g(G : D) - g_D(C, Pa_G(C)) + g_D(C, Pa_G(C) \setminus \{X\})$

*(k) Si el operador es D_ParentOfChild$(X, Y)$ entonces*
$g(G' : D) = g(G : D) - g_D(Y, Pa_G(Y)) + g_D(Y, Pa_G(Y) \setminus \{X\})$

*(l) Si el operador es D_HHOfChild*$(X, Y)$ *entonces*

$g(G' : D) = g(G : D) - g_D(Y, \{C, X\}) + g_D(Y, \{C\})$

**Demostración.** Los doce operadores definidos en el espacio de los C-RPDAGs, excepto D_ChildOfC($X$), son meras particularizaciones de los cinco operadores definidos por Acid y de Campos en [6] para RPDAGs. En efecto:

- A_ParentOfC($X$) =A_arc($X, C$),
- A_ChildOfC($X$)=A_arc($C, X$),
- A_ParentOfChild($X, Y$) = A_arc($X, Y$),
- A_SiblingOfC($X$) = A_link($C, X$),
- A_HHOfC($X, Y$) = A_hh($X, C, Y$),
- A_HHOfChild($X, Y$)=A_hh($X, Y, C$),
- D_ParentOfC($X$) =D_arc($X, C$),
- D_ParentOfChild($X, Y$)=D_arc($X, Y$),
- D_HHOfC($X, Y$) =D_arc($X, C$),
- D_HHOfChild($X, Y$)=D_arc($X, Y$) y
- D_SiblingOfC($X$) = D_link($C, X$).

Por lo tanto, podemos aplicar directamente los resultados correspondientes presentados en [6].

En relación con el operador D_ChildOfC($X$), implica eliminar no sólo el arco $C \rightarrow X$ sino todos los arcos en $G$ apuntando a $X$. Solamente la métrica local correspondiente a la variable $X$ debe, por lo tanto, ser modificada para reflejar que el conjunto de padres actual de $X$, $Pa_G(X)$ se ha transformado en $Pa_{G'}(X) = \emptyset$. Esta transformación da el resultado mostrado en (h). ∎

## 6.3. Resultados experimentales.

En esta sección se describen los experimentos realizados con el clasificador propuesto basado en C-RPDAGs, los resultados obtenidos, y un estudio comparativo con otros clasificadores bayesianos. Para ello hemos seleccionado 31 bases de datos bien conocidas, obtenidas del repositorio de aprendizaje automático *UCI repository of machine learning databases* [40], excepto las bases de datos *mofn-3-7-10* y *corral*, que fueron diseñadas por Kohavi [197]. Todos estos conjuntos de datos han sido ampliamente utilizados en la literatura especializada con fines comparativos bajo la perspectiva de la clasificación.

La tabla 6.2 muestra una breve descripción de las características de cada una de las bases de datos, incluyendo el número de casos (*instancias*), número de atributos (*atributos*) y el número de estados de la variable de clase (*clases)*. Estas bases de datos han sido preprocesadas, aquellas que tenían variables continuas, fueron discretizadas. La medida usada para discretizar fue la entropía con un número de intervalos variable, y se obtiene siguiendo el procedimiento de Fayyad e Irani [111]. Los casos con valores perdidos fueron omitidos.

Se ha llevado a cabo un estudio comparativo del método propuesto (abreviadamente, C-RPDAG) con otros clasificadores bayesianos, Los clasificadores estudiados han sido naïve bayes (NB), naïve bayes aumentado a estructura de árbol (TAN), naïve bayes aumentado a red bayesiana (BAN), una red bayesiana sin ninguna restricción estructural (UBN, del inglés *Unrestricted Bayesian Network*) y una segunda red bayesiana sin ninguna restricción estructural pero cuyo espacio de búsqueda no son DAGs sino RPDAGs (abreviadamente, lo denominaremos RPDAG).

Para construir los clasificadores BAN, UBN, C-RPDAG y RPDAG, se ha utilizado un enfoque de búsqueda+métrica. Para los clasificadores BAN y UBN, el algoritmo de búsqueda ha sido una búsqueda local en el espacio de los DAGs aumentados y de los DAGs sin restricciones, respectivamente (usando los operadores clásicos de inserción, eliminación e inversión de enlaces). En los clasificadores RPDAG y C-RPDAG, también se utiliza búsqueda local pero con los operadores adaptados a sus respectivos espacios de búsqueda.

La métrica usada para los anteriores clasificadores es la misma: la métrica BDeu [150], donde el parámetro que representa el tamaño muestral equiva-

| # | Base de datos | Instancias | Atributos | Clases |
|---|---|---|---|---|
| 1 | adult | 45222 | 14 | 2 |
| 2 | australian | 690 | 14 | 2 |
| 3 | breast | 682 | 10 | 2 |
| 4 | car | 1728 | 6 | 4 |
| 5 | chess | 3196 | 36 | 2 |
| 6 | cleve | 296 | 13 | 2 |
| 7 | corral | 128 | 6 | 2 |
| 8 | crx | 653 | 15 | 2 |
| 9 | diabetes | 768 | 8 | 2 |
| 10 | DNA-nominal | 3186 | 60 | 3 |
| 11 | flare | 1066 | 10 | 2 |
| 12 | german | 1000 | 20 | 2 |
| 13 | glass | 214 | 9 | 7 |
| 14 | glass2 | 163 | 9 | 2 |
| 15 | heart | 270 | 13 | 2 |
| 16 | hepatitis | 80 | 19 | 2 |
| 17 | iris | 150 | 4 | 3 |
| 18 | letter | 20000 | 16 | 26 |
| 19 | lymphography | 148 | 18 | 4 |
| 20 | mofn-3-7-10 | 1324 | 10 | 2 |
| 21 | mushroom | 8124 | 22 | 2 |
| 22 | nursery | 12960 | 8 | 5 |
| 23 | pima | 768 | 8 | 2 |
| 24 | satimage | 6435 | 36 | 6 |
| 25 | segment | 2310 | 19 | 7 |
| 26 | shuttle-small | 5800 | 9 | 7 |
| 27 | soybean-large | 562 | 35 | 19 |
| 28 | splice | 3190 | 60 | 3 |
| 29 | vehicle | 846 | 18 | 4 |
| 30 | vote | 435 | 16 | 2 |
| 31 | waveform-21 | 5000 | 21 | 3 |

Tabla 6.2: Descripción de los conjuntos de datos utilizados en nuestros experimentos.

lente es igual a 1 y se utiliza una estructura uniforme a priori. El punto de partida para la búsqueda local en los clasificadores UBN, RPDAG y C-RPDAG es un grafo vacío.

Una vez terminada la parte del aprendizaje estructural de los clasificadores, es necesario realizar un aprendizaje paramétrico. En todos los casos hemos usado un estimador suavizado basado en la *Ley de la sucesión de Laplace* [137]. Buscamos evitar los problemas de sobreajuste y falta de fiabilidad, que un estimador por máxima verosimilitud nos puede generar para conjuntos de datos con pequeñas muestras.

Los resultados de este estudio comparativo se pueden apreciar en la tabla 6.3. Se muestra la precisión de cada clasificador para cada conjunto de datos, junto con su desviación estándar. La precisión ha sido obtenida como la media de tres ejecuciones, la precisión en cada ejecución era calculada mediante validación cruzada de 10-hojas. Dentro de cada ejecución, cada una de las particiones de la base de datos (hojas) ha sido exactamente la misma para todos los clasificadores. Se ha repetido tres veces la validación cruzada para conseguir un buen balance entre el sesgo y la varianza de la estimación [194]. Los mejores y los peores resultados obtenidos para cada problema se han enfatizado usando letras en negrita e itálica, respectivamente. Además, en la parte de abajo de la tabla mostramos la precisión media y cuántas veces cada algoritmo ha sigo el mejor y cuántas el peor.

Podemos observar, que como era de esperar, ningún algoritmo es siempre mejor que el resto para todos los conjuntos de datos, y que cada una de las distintas aproximaciones pueden obtener el mejor resultado dependiendo del problema. No obstante, en general, los clasificadores RPDAG y C-RPDAG obtienen un mayor número de mejores resultados y no hay ningún caso donde sean el peor clasificador para un conjunto de datos.

Si observamos los problemas por el número de casos, se advierte que el clasificador C-RPDAG funciona bien en problemas con muchas instancias, como *nursery*, *mushroom* o *splice*, donde obtiene la mejor precisión, o en *letter* o *DNA-nominal* donde es el segundo mejor clasificador. También funciona bien en problemas con pocos casos como, por ejemplo, *corral*, *glass*, *glass2* (el mejor) o *iris* (el segundo mejor clasificador).

El clasificador presentado también tiene un buen comportamiento en problemas con un gran número de variables como es el caso de *splice* (el mejor) o *DNA-nominal* (el segundo) o en problemas con un gran número de clases como *glass*, *nursery* (el mejor) o *letter* (el segundo).

| # | NB | TAN | BAN | UBN | RPDAG | C-RPDAG |
|---|---|---|---|---|---|---|
| 1 | *83,13* (–) | 85,23 | **85,54** | 85,43 | 85,47 | 85,29 |
| 2 | 85,22 | *84,98* | 85,94 | 85,75 | **86,09** | 86,04 |
| 3 | 97,56 | *95,60* | 97,56 | **97,65** | 97,56 | 97,56 |
| 4 | *85,40* (–) | 93,86 | 93,09 | 92,98 | **93,87** | 93,07 |
| 5 | *87,84* (–) | 92,34 (–) | 96,88 | 97,02 | **97,53** | 96,53 |
| 6 | 83,24 | *80,89* | 82,57 | 83,57 | **84,03** | 81,78 |
| 7 | *86,82* (–) | 99,49 | **100,00** | **100,00** | **100,00** | **100,00** |
| 8 | 86,97 | *86,36* | 86,56 | 86,71 | 86,66 | **87,02** |
| 9 | *78,17* | 79,00 | 78,65 | 79,09 | **79,39** | 78,78 |
| 10 | 95,40 (–) | *94,82* (–) | 95,31 (–) | 95,95 | 95,96 | **96,15** |
| 11 | *80,40* | 82,90 | 82,93 | 82,27 | 82,36 | **83,27** |
| 12 | **75,27** | *72,80* | 74,67 | 73,80 | 74,57 | 74,30 |
| 13 | 72,78 | *69,05* | 71,83 | 69,06 | **73,84** | 73,10 |
| 14 | *83,38* | 85,04 | 85,04 | 84,82 | 84,61 | **85,86** |
| 15 | **83,46** | 82,84 | *82,10* | 83,33 | **83,46** | 82,22 |
| 16 | *85,00* | 90,42 | 88,75 | 89,58 | **94,58** | 88,75 |
| 17 | 94,22 | *93,56* | 94,44 | **94,89** | 94,00 | 94,67 |
| 18 | *74,01* (–) | **85,83** (+) | 84,35 (–) | 84,43 (–) | 84,69 | 85,08 |
| 19 | 84,41 | 81,94 | **84,43** | *79,08* | 82,43 | 81,54 |
| 20 | *85,40* (–) | 90,71 (–) | 87,29 (–) | 99,17 | **100,00** | 99,50 |
| 21 | *95,38* (–) | 99,98 | **100,00** | **100,00** | **100,00** | **100,00** |
| 22 | *90,26* (–) | 92,27 (–) | 91,82 (–) | 93,12 | 93,51 | **93,52** |
| 23 | *77,91* | 78,78 | 78,03 | 78,78 | **79,29** | 78,03 |
| 24 | *82,43* (–) | **88,47** | 88,29 | 83,49 (–) | 83,15 (–) | 88,19 |
| 25 | *92,16* (–) | **95,09** | 94,34 | 94,56 | 94,46 | 94,34 |
| 26 | *99,01* (–) | 99,66 | **99,72** | 99,69 | 99,69 | 99,60 |
| 27 | 91,70 | *86,05* | **92,53** | 88,55 | 89,85 | 90,33 |
| 28 | 95,45 (–) | *94,89* (–) | 95,28 | 95,73 | 95,78 | **96,27** |
| 29 | *62,85* (–) | 71,05 | 71,56 | 65,13 (–) | 64,97 (–) | **71,91** |
| 30 | *90,04* | 94,42 | 94,34 | **94,57** | 94,49 | 93,49 |
| 31 | 81,82 | **82,89** | 82,36 | *79,81* (–) | 79,96 (–) | 82,45 |
| Media | *85,39* | 87,46 | 87,94 | 87,68 | 88,19 | **88,34** |
| Mejor | 2 | 4 | 6 | 5 | 12 | 9 |
| Peor | 18 | 10 | 1 | 2 | 0 | 0 |

Tabla 6.3: Resultados experimentales. Precisión de cada clasificador para cada problema.

Si observamos detenidamente problemas como *corral*, *vehicle* o *letter*, podemos ver que se obtienen grandes diferencias respecto al clasificador naïve bayes, y en *mofn-3-7-10*, *glass* o *chess* con respecto al clasificador TAN.

Para confirmar los buenos resultados de nuestra propuesta, se ha utilizado el test de Wilcoxon de signos pareados por rangos para determinar si los resultados de cada propuesta son estadísticamente significativos. En la tabla 6.3 también se indica si un algoritmo es significativamente peor (-) o mejor (+) que el clasificador C-RPDAG, para cada conjunto de datos.

| | NB | TAN | BAN | UBN | RPDAG | C-RPDAG | Total mejor |
|---|---|---|---|---|---|---|---|
| NB | — | 12/**1** | 7/**0** | 7/**2** | 6/**1** | 5/**0** | 37/ **4** |
| TAN | 19/**12** | — | 12/**2** | 10/**4** | 7/**4** | 12/**1** | 60/**23** |
| BAN | 23/**11** | 18/**3** | — | 13/**4** | 11/**4** | 10/**0** | 75/**22** |
| UBN | 24/**10** | 20/**2** | 16/**1** | — | 8/**0** | 12/**0** | 80/**13** |
| RPDAG | 23/**11** | 24/**3** | 17/**2** | 20/**0** | — | 16/**0** | 100/**16** |
| C-RPDAG | 25/**14** | 19/**5** | 15/**4** | 17/**4** | 12/**3** | — | 88/**30** |
| Total peor | 114/**58** | 93/**14** | 67/**9** | 67/**14** | 44/**12** | 55/**1** | |

Tabla 6.4: Número de veces que el clasificador de la fila $i$ es mejor/**significativamente mejor** que el clasificador de la columna $j$.

En la tabla 7.6 comparamos cada clasificador con el resto. La entrada en la fila $i$, columna $j$ representa el número de veces que el clasificador $i$ es mejor/significativamente mejor (usando el test de Wilcoxon) que el clasificador $j$. De esta forma, para cada fila se muestra el número de veces que el clasificador correspondiente es mejor, mientras que para cada columna se muestra cuántas veces el clasificador correspondiente es peor. Observamos que el clasificador C-RPDAG puede competir favorablemente con el resto de clasificadores. Fijándonos sólo en los resultados significativos estadísticamente, el clasificador C-RPDAG obtiene un mayor número de mejores resultados y un número menor de peores resultados.

De estos resultados experimentales, un hallazgo inesperado ha sido el buen comportamiento que muestra el clasificador UBN, mejor que los clasificadores NB y TAN. Este hecho está aparentemente en contradicción con resultados anteriores [118]. Quizás la explicación pueda estar en el uso de una métrica diferente. Para determinar si diferentes métricas pueden generar precisiones de clasificación substancialmente diferentes, hemos repetido los mismos experimentos previos con el clasificador C-RPDAG, pero usando la

métrica BIC en lugar de la métrica BDeu. Los resultados de la comparación entre C-RPDAG$_{BIC}$ y C-RPDAG$_{BDeu}$ se muestran en la tabla 6.5. Diferencias significativas (también usando el test de Wilcoxon) se han marcado en la tabla con un asterisco. Estos resultados indican que la versión que usa la métrica BIC tiene un peor comportamiento en clasificación, que la que utiliza la métrica BDeu. Esto ilustra nuestra premisa, la elección de una métrica u otra es relevante para tareas de clasificación supervisada.

Además creemos que el hecho de utilizar un aprendizaje paramétrico basado en *La Ley de la sucesión de Laplace*, también influye en la mejora de los resultados en clasificación, mientras que, como contrapartida, Friedman y col. en [118] la estimación paramétrica se realiza por máxima verosimilitud, sin ningún tipo de corrección para muestras pequeñas.

Otro aspecto importante a tener en cuenta, además de la precisión, es la eficiencia. Aunque el tiempo es una medida poco precisa de la eficiencia, se han medido los tiempos de ejecución para todos los algoritmos, excluyendo al naïve bayes, el cual, obviamente es mucho más rápido que el resto. En la tabla 6.6 se muestran los tiempos de cada algoritmo en relación con el tiempo consumido por el clasificador C-RPDAG (es decir, $\frac{ttiempoClasficador_i}{tiempoC-RPDAG}$). Estos tiempos se han calculado haciendo la media para todos los conjuntos de datos y para los conjuntos de datos más grandes (aquellos que incluyen al menos 20 atributos). Como se esperaba, el clasificador C-RPDAG es más rápido que los clasificadores BAN y UBN. Diferencia, que se incrementa en conjuntos de datos más grandes, y en ese caso el clasificador C-RPDAG es incluso más rápido que el clasificador TAN. No obstante, sorprendentemente, RPDAG es dos veces más rápido que el clasificador C-RPDAG (pensamos que esto se debe a una implementación más fina del clasificador RPDAG).

## 6.4. Discusión.

Hasta ahora, se pensaba que un enfoque métrica+búsqueda podía construir redes bayesianas que maximizaran el valor de la métrica pero que se comportaban pobremente como clasificadores. Esto es una verdad a medias, debido a que el uso de un algoritmo de búsqueda especializado en clasificación nos permite obtener mejores resultados que otros algoritmos de búsqueda no orientados a clasificación, aún con enfoque métrica+búsqueda. De hecho, se

| Conj. de datos | C-RPDAG$_{BIC}$ | C-RPDAG$_{BDeu}$ |
|---|---|---|
| adult | **85,42 ± 0,48**∗ | 85,29 ± 0,45 |
| australian | **86,28 ± 3,57** | 86,04 ± 4,25 |
| breast | **97,56 ± 1,71** | **97,56 ± 1,71** |
| car | 85,63 ± 2,69 | **93,07 ± 1,97**∗ |
| chess | 95,81 ± 1,24 | **96,53 ± 1,19**∗ |
| cleve | **82,46 ± 7,61** | 81,78 ± 7,56 |
| corral | **100,00 ± 0,00** | **100,00 ± 0,00** |
| crx | 86,62 ± 3,67 | **87,02 ± 3,70** |
| diabetes | 78,57 ± 3,44 | **78,78 ± 3,74** |
| DNA-nominal | **96,33 ± 1,04**∗ | 96,15 ± 1,10 |
| flare | 82,77 ± 2,83 | **83,27 ± 2,98** |
| german | **74,40 ± 4,46** | 74,30 ± 4,07 |
| glass | 70,12 ± 7,26 | **73,10 ± 7,51** |
| glass2 | 84,83 ± 8,42 | **85,86 ± 7,81** |
| heart | **82,59 ± 8,25** | 82,22 ± 8,16 |
| hepatitis | 87,50 ± 10,11 | **88,75 ± 9,47** |
| iris | 94,22 ± 4,77 | **94,67 ± 4,64** |
| letter | 76,73 ± 0,95 | **85,08 ± 0,75**∗ |
| lymphography | **81,78 ± 10,56** | 81,54 ± 11,07 |
| mofn-3-7-10 | 93,56 ± 1,88 | **99,50 ± 1,15**∗ |
| mushroom | **100,00 ± 0,00** | **100,00 ± 0,00** |
| nursery | 91,30 ± 0,76 | **93,52 ± 0,61**∗ |
| pima | **78,51 ± 4,51** | 78,03 ± 4,54 |
| satimage | 84,57 ± 1,12 | **88,19 ± 1,12**∗ |
| segment | 92,17 ± 1,69 | **94,34 ± 1,29**∗ |
| shuttle-small | **99,79 ± 0,17**∗ | 99,60 ± 0,23 |
| soybean-large | **91,70 ± 3,56** | 90,33 ± 3,25 |
| splice | **96,30 ± 1,05** | 96,27 ± 1,13 |
| vehicle | 71,75 ± 3,80 | **71,91 ± 3,91** |
| vote | 92,96 ± 4,17 | **93,49 ± 3,22** |
| waveform-21 | **82,47 ± 1,58** | 82,45 ± 1,67 |
| Total mejor | 12 | 16 |
| Total signific. mejor | 3 | 7 |

Tabla 6.5: Clasificador C-RPDAG usando las métricas BIC y BDeu.

| | TAN | BAN | UBN | RPDAG |
|---|---|---|---|---|
| Todos los conj. de datos | 0,539 | 1,081 | 1,192 | 0,546 |
| Conj. de datos grandes | 1,155 | 1,306 | 1,383 | 0,567 |

Tabla 6.6: Tiempos de ejecución medios consumidos por cada algoritmo en relación con el clasificador C-RPDAG.

ha desarrollado un nuevo método basado en el paradigma métrica+búsqueda para aprender clasificadores bayesianos sin restricciones estructurales. Este

nuevo método está basado en el uso de C-RPDAGs, un tipo de grafos acíclicos parcialmente dirigidos, como elementos del espacio de búsqueda. Esto grafos combinan las ideas de equivalencia en clasificación y equivalencia en independencia, para producir un espacio de búsqueda más reducido, focalizado y robusto. Los resultados experimentales muestran que el método propuesto basado en C-RPDAGs puede obtener, de manera eficiente, clasificadores que pueden competir favorablemente con otros clasificadores bayesianos del estado del arte.

# Capítulo 7

# Aplicaciones al análisis de datos de expresión genética.

Los datos de expresión genética nos permiten estudiar de una forma rápida y directa miles de genes simultáneamente, viendo qué genes están expresados y cuáles no. Al poder estudiar tantos genes a la vez, podremos conseguir un mayor entendimiento de procesos biológicos como el cáncer, y la identificación de los genes implicados en los mismos.

En la clasificación supervisada se tiene la tarea de asociar los valores que toman un conjunto de variables con un conjunto discreto de valores, denominados clases. Por tanto, un clasificador se puede ver como una función que a partir de una instancia nos dice la clase a la cual pertenece. En esta memoria, se han presentado diferentes clasificadores que en este capítulo se aplicarán en la clasificación de datos de expresión genética. Es decir, predecir una clase (que, como veremos, serán distintos tipos de cáncer o supervivencia al mismo) en base a los niveles de expresión de un conjunto de genes.

En esta memoria se ha mostrado una nueva metodología para incorporar conocimiento experto en el aprendizaje automático de redes bayesianas a partir de datos, mediante el uso de restricciones estructurales. En el análisis de datos de expresión genética, se ha intentando incorporar conocimiento adicional que ayude en el aprendizaje de las redes. En este capítulo mostraremos un estudio experimental de cómo puede ser la incorporación de conocimiento experto al aprendizaje estructural de una red bayesiana con datos provenientes de microarrays de ADN.

## 7.1. Clasificación bayesiana de datos de expresión genética.

Los clasificadores que vamos a utilizar para analizar datos de expresión genética serán los presentados en esta memoria, concretamente en los capítulos 4, 5 y 6. Por tanto, tenemos: árboles de clasificación que usan una estimación bayesiana, multiredes bayesianas y el clasificador C-RPDAG.

A diferencia de las redes bayesianas, las multiredes nos permiten representar independencias asimétricas. Por este motivo, creemos que puede ser interesante su aplicación a datos de expresión genética, pues creemos que al poder representar este tipo de independencias, pueden representar de forma efectiva contextos celulares [307]: un gen se puede comportar de forma distinta dependiendo del contexto en que esté la célula (por ejemplo, que la célula sea cancerígena o no).

Los árboles de clasificación con una estimación bayesiana también se han incluido en este estudio experimental puesto que nos parece interesante hacerlo por dos motivos: por un lado, nos va a permitir comparar los clasificadores bayesianos con un árbol de clasificación y, por otro lado, podemos ver el efecto de la partición recursiva de los datos en los árboles de clasificación. Este problema probablemente se verá agravado por la poca cantidad de instancias que presentan los datos de expresión genética.

Finalmente, también consideramos conveniente incluir un clasificador bayesiano sin restricciones estructurales pero orientado a clasificación supervisada como es el clasificador C-RPDAG. El hecho de usar un clasificador con mayor poder expresivo (pues no tiene restricciones estructurales) que los clasificadores bayesianos habituales usados en la literatura, nos parece especialmente indicado en datos que sufren de la maldición de la dimensionalidad. Además nos permite, por un lado, centrar la búsqueda desde la perspectiva de la clasificación y, por otro lado, explorar un espacio de búsqueda más robusto y reducido.

### 7.1.1. Conjuntos de datos.

Para hacer el estudio experimental de los distintos clasificadores se ha escogido una serie de bases de datos de expresión genética obtenido a partir de muestras de tejidos con distintos tipos de cáncer. La tabla 7.1 muestra una breve descripción de las características de estas bases de datos: la columna *Tamaño* dice el número de casos, *Genes* nos da el número de genes medidos en los microarrays de ADN y, finalmente, la columna *Clases* muestra el número de estados distintos de la variable clase.

| Conj. de datos | Tamaño | Genes | Clases |
|---|---|---|---|
| Breast-cancer | 97 | 24481 | 2 |
| CNS | 60 | 7129 | 2 |
| Colon-cancer | 62 | 2000 | 2 |
| DLBCL-MIT | 77 | 7129 | 2 |
| DLBCL-Stanford | 47 | 4026 | 2 |
| Leukemia ALL-AML | 72 | 7129 | 2 |
| Leukemia-MLL | 72 | 12582 | 3 |
| Lung-cancer | 203 | 12600 | 5 |
| Prostate-tumor | 136 | 12600 | 2 |

Tabla 7.1: Conjuntos de datos de expresión genética para distintos tipos de cáncer utilizados en los experimentos.

El primer conjunto de datos trata sobre la recaída de los pacientes con cáncer de mama (en la tabla 7.1 se ha notado como *Breast*) y presentado por Van't Veer y col. en [344][1]. En esta base de datos, se tiene para el conjunto de entrenamiento 78 muestras de pacientes, donde 34 de ellos han tenido una recaída en los primeros cinco año. Los 44 restantes se han mantenido sanos después del diagnóstico inicial, durante cinco años. En el conjunto de test se tienen 12 pacientes que han sufrido una recaída y 7 que no.

El conjunto sobre el tumor embrionario del sistema nervioso central (*CNS* en la tabla 7.1) fue presentado por Pomeroy y col. en [275][2]. De dicho trabajo

[1]Disponible en http://www.rii.com/publications/2002/vantveer.htm
[2]Disponible en http://www-genome.wi.mit.edu/mpr/CNS/

sólo se ha cogido el conjunto de datos C. En este caso se tienen 60 muestras de pacientes, de las que 21 han sobrevivido después del tratamiento mientras que hay 39 casos que no lo han hecho.

Alon y col. presentan en [12] un estudio sobre el cáncer de colon[3]. En este conjunto de datos (*Colon-cancer* en la tabla 7.1) tenemos biopsias sobre tumores de colon, donde 40 representan muestras de tejido de los tumores malignos y 22 muestras pertenecen a muestras sanas del colon de los mismos pacientes.

El conjunto de datos de Alizadeh y col. [9] sobre el linfoma[4] difuso de células B grandes (*DLBCL-Stanford* en la tabla 7.1) es, como vimos, una de las bases de datos más utilizadas en la literatura[5]. Contiene dos tipos distintos de linfoma: 24 muestras con linfoma del tipo centro germinal B y 23 muestras del tipo célula B activada.

Otro conjunto de datos sobre el linfoma difuso de células B grandes, es el conjunto de datos de expresión genética presentado por Shipp y col. en [314][6] (*DLBCL-MIT* en la tabla 7.1). Presentan dos problemas de clasificación, uno de distinción de tipos de linfomas y un segundo sobre predicción de supervivencia de los pacientes. En los experimentos se ha utilizado el primero, donde tenemos 58 muestras de un tipo de linfoma y 19 muestras de otro tipo.

El conjunto sobre leucemia[7] presentado por Golub y col. en [136][8] es, como dijimos, el más usado en clasificación supervisada de datos de expresión genética (*Leukemia ALL-AML* en la tabla 7.1). Las distintas muestras están clasificadas según el tipo de leucemia que sufre el paciente: 25 pacientes

[3]Disponible en http://microarray.princeton.edu/oncology/affydata/index.html

[4]Los linfomas son un conjunto de enfermedades cancerosas que se desarrollan en el sistema linfático, que también forman parte del sistema inmunológico del cuerpo humano. A los linfomas también se les llama los tumores sólidos hematológicos para diferenciarlos de las leucemias.

[5]Disponible en http://llmpp.nih.gov/lymphoma/

[6]Disponible en http://www-genome.wi.mit.edu/cgi-bin/cancer/datasets.cgi

[7]La leucemia es un grupo de enfermedades malignas de la médula ósea (cáncer hematológico) que provoca un aumento incontrolado de glóbulos blancos en la médula ósea, que suelen pasar a la sangre aunque en ocasiones no lo hacen. Ciertas proliferaciones malignas de glóbulos rojos se incluyen entre las leucemias.

[8]Disponible en http://www.broad.mit.edu/mpr/data_set_ALL_AML.html

con leucemia mieloide aguda y 47 pacientes con leucemia linfoide aguda.

También sobre leucemia, pero considerando un tipo adicional, es el conjunto de datos presentado por Armstrong y col. en [18][9] (*Leukemia-MLL* en la tabla 7.1), donde tenemos 28 muestras de pacientes con leucemia mieloide aguda, 24 pacientes con leucemia linfoide aguda y 20 muestras de pacientes con leucemia de linaje mixto.

Sobre el cáncer de pulmón se tiene otro conjunto de datos, presentado por Bhattacharjee y col. en [38][10] (*Lung-cancer* en la tabla 7.1), donde tenemos 203 muestras con 4 tipos distintos de cáncer de pulmón y muestras de pulmón sano.

Finalmente, tenemos un conjunto de datos sobre el cáncer de próstata[11] extraído del trabajo de Singh y col. [316] (*Prostate-tumor* en la tabla 7.1). En este trabajo se presentan dos problemas de clasificación: uno de distinción entre distintos tipos de tumor y otro sobre las tasas de supervivencia. Nosotros nos hemos centrado en el primero, donde tenemos 52 muestras con tumor de próstata y 50 casos con muestras sanas de próstata. Estos datos se han complementado con un conjunto de test[12], que contenía más genes, pero estos genes adicionales se han eliminado.

Además de en sus respectivos sitios de origen, todas las bases de datos se pueden encontrar en el repositorio *Kent Ridge Bio-medical Dataset*[13] [217], donde almacenan las bases de datos en el formato de UCI[40] o MLC++ [198] y en el formato *arff* de WEKA [359]. De este repositorio hemos obtenido estas bases de datos y no de sus sitios originales.

Estas bases de datos han sido ampliamente utilizadas dada su disponibilidad de forma pública y, por tanto, distintos clasificadores se han probado sobre dichos conjuntos de datos. En la tabla 7.2 se pueden observar los mejores resultados en precisión que se han encontrado en la literatura para estas bases de datos [237, 254, 42, 227, 61, 62, 60].

[9]Disponible en http://www.broad.mit.edu/publications/broad901
[10]Disponible en http://www-genome.wi.mit.edu/mpr/lung/
[11]Disponible en http://www.broad.mit.edu/publications/broad895
[12]Disponible en http://carrier.gnf.org/welsh/prostate/
[13]Disponible en http://datam.i2r.a-star.edu.sg/datasets/krbd/

| Conj. de datos | Mejor Precisión |
|---|---|
| Breast-cancer | 89,47 |
| CNS | 86,5 |
| Colon-cancer | 98,5 |
| DLBCL-MIT | 98,2 |
| DLBCL-Stanford | 97,6 |
| Leukemia ALL-AML | 98,8 |
| Leukemia-MLL | 100 |
| Lung-cancer | 93,5 |
| Prostate-tumor | 98,8 |

Tabla 7.2: Mejores resultados obtenidos para los conjuntos de datos de expresión genética utilizadas en los experimentos.

### 7.1.2. Preprocesamiento.

El primer paso en el procesamiento de las distintas bases de datos, ha sido unir aquellas que se dividían en conjuntos de entrenamiento y test. Nos parece más acertado unir ambos conjuntos y utilizar validación cruzada para estimar la precisión de los clasificadores. Los problemas que se componían de conjuntos de entrenamiento y test eran: *Breast-cancer*, *DLBCL-MIT*, *Leukemia ALL-AML*, *Leukemia-MLL* y *Prostate-tumor*.

**Imputación de datos perdidos/desconocidos.** Los distintos conjuntos de datos no tenían datos perdidos/desconocidos a excepción de la base de datos DLBCL-Stanford. Para poder trabajar con este conjunto de datos se ha utilizado el algoritmo de imputación de datos por los mínimos cuadráticos [185] denominado *LLSImpute* (del inglés, *local least squares imputation method*). Este método de imputación de datos perdidos/desconocidos representa un gen objetivo (un gen con datos perdidos) como combinación lineal de genes similares. Los genes similares se escogen mediante el algoritmo de clustering de los k vecinos más cercanos [114].

La implementación usada de dicho algoritmo es la que se encuentra en el

paquete *pcaMethods*[14] [324] para el lenguaje estadístico R[15] [280]. Los ficheros utilizados han sido leídos en formato .arff con el paquete RWeka [156].

**Discretización.** Como vimos, otro de los problemas que presentaban los datos provenientes de microarrays de ADN era el hecho de que los niveles de expresión se movían en un rango de valores continuos. La mayoría de los autores optan por una discretización en tres estados[16]. La discretización tiene la ventaja de ser más estable frente a las variaciones aleatorias o sistemáticas que se producen en los niveles de expresión de los microarrays. Pero tiene el inconveniente de que puede producirse una pérdida de información. Por ello, algunos autores han propuesto la utilización de modelos basados en gaussianas, mixturas de gaussianas o regresión no paramétrica.

Se han escogido dos tipos de discretizaciones para tratar los datos. En primer lugar se ha escogido la discretización usada por Friedman y col. en [122]. Esta es la discretización más usada en la literatura cuando se trabaja con datos de expresión genética. En este caso la discretización se hace en tres categorías: infraexpresados, normal y sobreexpresados, dependiendo si el nivel de expresión es significativamente menor, similar o mayor que un nivel de control, respectivamente. El nivel de control de un gen se puede determinar experimentalmente [94] o puede fijarse como el nivel medio de expresión del gen en los distintos experimentos. En [122] se fijó un valor umbral respecto a la razón entre la medida y el valor de control, en este caso, la media. El umbral elegido fue de 0,5 en escala logarítmica (base 2). De este modo, valores con un índice de control menores que $2^{-0,5}$ se consideran infraexpresados y valores mayores que $2^{0,5}$ se consideran sobreexpresados.

El segundo método de discretización escogido ha sido el propuesto por Fayyad e Irani en [111] y que ha sido ampliamente utilizado en problemas de

[14] Disponible en http://www.bioconductor.org/packages/release/bioc/html/pcaMethods.html .

[15] Disponible en http://www.r-project.org/ .

[16] Téngase en cuenta que el modelo de red de regulación genética que se utiliza hoy en día está basado en el modelo de Kauffman [180] donde los genes sólo tienen dos estados, encendido o apagado. A pesar de ser un modelo simplificado ha resultado ser bastante útil desde el punto de vista predictivo, permitiendo deducir resultados biológicamente relevantes.

clasificación supervisada. Este método selecciona recursivamente los puntos de corte mediante un algoritmo de minimización de la entropía entre cada atributo y la clase. Con este método, en los datos de expresión genética con los que estamos trabajando, la mayoría de los genes son discretizados en dos estados, si bien, en algunos casos se discretizan en tres estados. También es necesario notar que mediante esta discretización aparecen variables discretizadas a un sólo estado, es decir, son irrelevantes.

Para quedarnos con uno de los dos métodos de discretización propuesto (el más usado en datos de expresión genética o el más usado en problemas de clasificación supervisada), se ha realizado un estudio previo. Hemos escogido el clasificador naïve bayes debido a su rapidez y a que está incluido en algunos de los clasificadores que presentaremos (esto es, multiredes bayesianas y C-RPDAG). Se ha realizado una validación cruzada de 20 hojas. De esta forma hemos comparado los resultados obtenidos por los distintos métodos de discretización tal y como se puede observar en la tabla 7.3. Los resultados que obtenemos con la discretización de Friedman y col. son relativamente malos, no obstante, pensamos que se debe a que éste método mantiene variables irrelevantes que sí son eliminadas por el otro método (recordemos que el clasificador naïve bayes es sensible a variables irrelevantes).

| Conj. de datos | Fayyad e Irani | Friedman y col. |
|---|---|---|
| Breast-cancer | 90,7216 | 67,0103 |
| CNS | 93,3333 | 61,6667 |
| Colon-cancer | 93,5484 | 67,7419 |
| DLBCL-MIT | 92,2078 | 77,9221 |
| DLBCL-Stanford | 100,00 | 91,4894 |
| Leukemia ALL-AML | 98,6111 | 86,1111 |
| Leukemia-MLL | 98,6111 | 91,6667 |
| Lung-cancer | 78,3251 | 61,0837 |
| Prostate-tumor | 79,4118 | 55,1471 |

Tabla 7.3: Precisión del clasificador naïve bayes para cada problema discretizado de dos formas, usando validación cruzada de 20 hojas.

Debemos mencionar que esta fase de discretización y la posterior fase de selección de genes (que ahora veremos), han sido realizadas con el entorno

WEKA[17] [359] (del inglés *Waikato environment for knowledge analysis*).

**Selección de genes.** Posteriormente a la discretización de los datos se ha realizado una fase de selección de genes para reducir la dimensionalidad de estos problemas, ya que algunos de los clasificadores propuestos no serían capaces de tratar con tantas variables en un tiempo razonable. En esta fase de selección de características se ha realizado un proceso similar al propuesto por Armañanzas y col. en [14]. Primero, en la discretización de Fayyad e Irani, se han eliminado aquellas variables que habían sido discretizadas a un solo estado. En una segunda etapa se ha realizado una selección de genes (Armañanzas y col. los denominaban genes prototipos) donde se han escogido aquellos genes representativos de un grupo de genes que se coexpresan a la vez. En una tercera etapa se ha realizado una selección de genes mediante un ranking de las variables, con el objetivo de reducir el número de genes a tratar.

En la etapa de selección de genes representativos, se ha llevado a cabo la selección de variables CFS [141] (del inglés, *correlation based filter*). Esta selección de variables está basada en la correlación de los genes con la variable a clasificar. Trata de encontrar el subconjunto óptimo de atributos altamente correlados con la clase y, al mismo tiempo, con un bajo grado de redundancia entre ellos. Para ello, busca un subconjunto de atributos considerando la capacidad predictora de cada uno individualmente, pero también se busca que haya poca correlación entre los atributos.

Obsérvese que en un primer paso se han eliminado genes irrelevantes y en un segundo paso además de genes poco relevantes, se han eliminado genes redundantes. En los datos de expresión genética tendremos genes que son independientes del proceso biológico que estemos estudiando (genes irrelevantes). Por ejemplo, en nuestros experimentos, está claro que no todos los genes están involucrados en los distintos tipos de cánceres que estamos estudiando. Por otro lado, es común en los datos de expresión genética, que haya genes que se expresen a la vez. Dichos genes nos van a dar la misma información sobre la variable a clasificar. Por tanto, la eliminación de los genes redundantes nos permite no considerar genes coexpresados y quedarnos con genes representativos.

[17]Disponible en http://www.cs.waikato.ac.nz/ml/weka/ .

En la tabla 7.4 podemos ver, una nueva comparativa entre la discretización de Fayyad e Irani frente a la discretización de Friedman y colaboradores. Se ha vuelto a usar validación cruzada de 20 hojas y un clasificador naïve bayes. En la tabla también aparecen las variables que quedan en cada caso tras aplicar la selección de variables CFS. La idea de esta comparativa es volver a comprobar que para estos problemas, es mejor usar la discretización de Fayyad e Irani que la de Friedman y colaboradores. Téngase en cuenta que la segunda, al no detectar valores irrelevantes, podía darnos problemas con el clasificador naïve bayes tal y como comentamos al ver la tabla 7.3. Ahora, después de hacer una selección de genes, los resultados deben ser más realistas.

| Conj. de datos | Fayyad e Irani | Genes | Friedman y col. | Genes |
|---|---|---|---|---|
| Breast-cancer | 100,0 | 143 | 95,88 | 86 |
| CNS | 96,67 | 39 | 100,0 | 41 |
| Colon-cancer | 98,39 | 27 | 90,32 | 18 |
| DLBCL-MIT | 100,0 | 93 | 98,70 | 50 |
| DLBCL-Stanford | 100,0 | 71 | 100,0 | 31 |
| Leukemia ALL-AML | 100,0 | 82 | 100,0 | 41 |
| Leukemia-MLL | 100,0 | 149 | 100,0 | 110 |
| Lung-cancer | 99,02 | 551 | 97,04 | 577 |
| Prostate-tumor | 97,79 | 75 | 91,91 | 37 |

Tabla 7.4: Precisión del clasificador naïve bayes usando validación cruzada de 20 hojas para los dos algoritmos de discretización. A los conjuntos de datos se les ha aplicado una selección de variables CFS posterior a la discretización.

Se puede observar que después de la reducción de genes, ambos métodos de discretización nos dan resultados bastante buenos (incluso superan en algunos casos a los mejores resultados encontrados en la literatura y expuestos en la tabla 7.1). Si bien, los resultados obtenidos con el método de Fayyad e Irani siguen siendo iguales o mejores a los del método de Friedman y col. (con la sola excepción del conjunto de datos *CNS*). Por tanto, creemos que la discretización que habitualmente se aplica en problemas de clasificación (la propuesta por Fayyad e Irani) también se puede aplicar a datos de expresión genética. Utilizaremos sólo este tipo de discretización de aquí en adelante.

Si nos fijamos en la tabla 7.4, después de la reducción de genes en la discretización y la selección de genes posterior, el número de genes es bastante elevado en algunos casos. Por este motivo, consideramos necesario hacer una segunda reducción de la dimensionalidad de los datos. El objetivo es que los clasificadores que vamos a usar puedan construirse en un tiempo razonable. Es por este motivo que vamos a introducir una tercera etapa en la selección de genes.

En este caso nos vamos a limitar a reducir aquellas bases de datos que tengan más de 100 genes. Se va a utilizar un ranking de variables y nos quedaremos con las 100 variables más relevantes (por ello no tiene sentido aplicarlo a aquellos conjuntos que tengan menos variables). El criterio usado para ordenar las variables ha sido la *ganancia de información*, que ya hemos visto en los capítulos 4 y 5 y que fue definida por Quinlan para el algoritmo ID3 [279]. Para una variable $X$ y conjunto de datos $D$:

$$ganancia(C|X) = H_D(C) - H_D(C|X)$$

donde $H_D(C)$ es la entropía de la clase y $H_D(C|X)$ es la entropía de la clase condicionada a $X$.

### 7.1.3. Resultados experimentales.

En esta sección se van a utilizar las bases de datos de expresión genética de la tabla 7.1 procesados en la forma que se ha detallado. Los clasificadores que se van a utilizar en los experimentos son: árboles de clasificación usando una estimación bayesiana (que denotaremos en las tablas como $CT_D$), el clasificador C-RPDAG y multiredes bayesianas. Además, con el objetivo de realizar comparaciones, se van a utilizar dos clasificadores bayesianos clásicos: naïve bayes ($NB$) y naïve bayes aumentado a estructura de árbol ($TAN$).

Todos los resultados de esta sección se han llevado a cabo con la herramienta *Elvira*[18] [105].

La métrica usada para el clasificador C-RPDAG es la métrica BDeu [150], donde el parámetro que representa el tamaño muestral equivalente es igual a 1 y se utiliza una estructura uniforme a priori. El punto de partida para la

[18]Disponible en http://leo.ugr.es/elvira/

búsqueda local en el clasificador C-RPDAG es un grafo vacío. Para el aprendizaje paramétrico de los clasificadores naïve bayes, TAN y C-RPDAG se ha usado un estimador suavizado basado en la *Ley de la sucesión de Laplace* [137]. Buscamos evitar los problemas de sobreajuste y falta de fiabilidad, que un estimador por máxima verosimilitud nos puede generar para conjuntos de datos con pequeñas muestras.

Dentro de las multiredes bayesianas hemos utilizado las dos multiredes de filtrado propuestas en la discusión del capítulo 5, esto es, la multired que usaba la heurística *atributo padre del resto* con la métrica K2 ($BMN_{f5}$) y la multired que utilizaba el *filtro Bhattacharyya* ($BMN_{f12}$). También se ha considerado la multired bayesiana de envoltura ($BMN_w$). En las hojas de las multiredes utilizaremos el clasificador naïve bayes.

Los resultados de este estudio comparativo se pueden apreciar en la tabla 7.5. Se muestra la precisión de cada clasificador para cada conjunto de datos. La precisión ha sido obtenida mediante validación cruzada dejar-uno-fuera.

| Problema | NB | TAN | $CT_D$ | C-RPDAG | $BMN_w$ | $BMN_{f5}$ | $BMN_{f12}$ |
|---|---|---|---|---|---|---|---|
| Breast-cancer | 100, 0 | 98, 97 | 77, 32 | 100, 0 | 98, 97 | 98, 97 | 98, 97 |
| CNS | 96, 67 | 98, 33 | 85, 00 | 100, 0 | 96, 67 | 93, 33 | 98, 33 |
| Colon-cancer | 98, 39 | 98, 39 | 93, 55 | 100, 0 | 98, 39 | 98, 39 | 98, 39 |
| DLBCL-MIT | 100, 0 | 100, 0 | 94, 81 | 100, 0 | 100, 0 | 100, 0 | 100, 0 |
| DLBCL-Stanford | 100, 0 | 100, 0 | 93, 62 | 100, 0 | 100, 0 | 100, 0 | 100, 0 |
| Leukemia ALL-AML | 100, 0 | 100, 0 | 88, 89 | 100, 0 | 100, 0 | 100, 0 | 100, 0 |
| Leukemia-MLL | 100, 0 | 100, 0 | 97, 22 | 100, 0 | 100, 0 | 97, 22 | 100, 0 |
| Lung-cancer | 98, 52 | 97, 54 | 87, 69 | 98, 52 | 97, 54 | 97, 54 | 98, 52 |
| Prostate-tumor | 97, 79 | 97, 06 | 90, 44 | 100, 0 | 97, 79 | 97, 06 | 97, 06 |
| **Medias** | 99, 04 | 98, 92 | 89, 84 | 99, 84 | 98, 82 | 98, 06 | 99, 03 |

Tabla 7.5: Resultados experimentales. Precisión de cada clasificador para cada problema usando validación cruzada dejar-uno-fuera.

En la tabla 7.6 comparamos cada clasificador con el resto. La entrada en la fila $i$, columna $j$ representa el número de veces que el clasificador $i$ es mejor/significativamente mejor que el clasificador $j$. De esta forma, para cada fila se muestra el número de veces que el clasificador correspondiente es

mejor, mientras que para cada columna se muestra cuantas veces el clasificador correspondiente es peor. Se ha utilizado el test de Wilcoxon de signos pareados por rangos para determinar si los resultados de cada propuesta son estadísticamente significativos.

| | NB | TAN | $CT_D$ | C-RPDAG | $BMN_w$ | $BMN_{f5}$ | $BMN_{f12}$ | Total mejor |
|---|---|---|---|---|---|---|---|---|
| NB | — | 3/**0** | 9/**2** | 0/**0** | 2/**0** | 5/**0** | 2/**0** | 21/ **0** |
| TAN | 1/**0** | — | 9/**2** | 0/**0** | 3/**0** | 2/**0** | 0/**0** | 15/ **0** |
| $CT_D$ | 0/**0** | 0/**0** | — | 0/**0** | 0/**0** | 0/**0** | 0/**0** | 0/ **0** |
| C-RPDAG | 3/**0** | 5/**0** | 9/**2** | — | 5/**0** | 6/**0** | 4/**0** | 32/ **0** |
| $BMN_w$ | 0/**0** | 1/**0** | 9/**2** | 0/**0** | — | 3/**0** | 1/**0** | 14/ **0** |
| $BMN_{f5}$ | 0/**0** | 0/**0** | 8/**2** | 0/**0** | 3/**0** | — | 0/**0** | 11/ **0** |
| $BMN_{f12}$ | 1/**0** | 1/**0** | 9/**2** | 0/**0** | 0/**0** | 3/**0** | — | 14/ **0** |
| Total peor | 5/**0** | 10/**0** | 53/**12** | 0/**0** | 13/**0** | 19/**0** | 7/**0** | |

Tabla 7.6: Número de veces que el clasificador de la fila $i$ es mejor/**significativamente mejor** que el clasificador de la columna $j$.

Observando las tablas 7.5 y 7.6, podemos ver con claridad que el clasificador C-RPDAG obtiene mejores resultados que el resto de clasificadores. Nos puede llamar la atención que el clasificador naïve bayes obtenga mejores resultados que el clasificador TAN. Esto creemos que es debido a que se ha usado como guía en los distintos métodos de preprocesamiento que se han estudiado. Además, tengamos en cuenta, que la selección de genes representativos que se ha realizado, beneficia al clasificador naïve bayes, puesto que nos devuelve genes altamente correlados con la clase e independientes entre sí.

Al proponer el uso de las multiredes bayesianas en el tratamiento de datos de expresión genética pensamos que podrían aprovecharse de las independencias asimétricas que podemos encontrar en los contextos celulares. No obstante, aún usando un sólo nivel en la estructura del árbol, las multiredes sufren del problema del particionamiento de los datos, problema que se agudiza ante la presencia de tan pocos casos. Los malos resultados obtenidos por el árbol de clasificación usando una estimación bayesiana profundizan en esta idea.

## 7.2. Uso de conocimiento experto en datos de expresión genética.

En este apartado estudiaremos el efecto de la utilización de conocimiento experto en el aprendizaje de redes bayesianas a partir de datos de expresión genética. Para ello, nos basaremos en que las redes bayesianas son visualmente interpretables, por tanto, trataremos de incorporar conocimiento experto de forma gráfica y en consonancia con la interpretabilidad del modelo, mediante restricciones estructurales.

Las restricciones estructurales que vamos a utilizar, son las vistas en el capítulo 3. Así pues, vamos a considerar tres tipos de restricciones: (1) existencia de arcos y aristas, (2) ausencia de arcos y aristas, y (3) restricciones de orden. Todas ellas son consideradas restricciones fuertes, en el sentido de que se asumen ciertas para cada red bayesiana representando el dominio de conocimiento, y por lo tanto todas las redes bayesianas candidatas deben cumplirlas.

### 7.2.1. Conjunto de datos de microarrays de ADN.

Vamos a estudiar el uso de restricciones estructurales como método de incorporación de conocimiento experto con datos de expresión genética de la levadura *Saccharomyces cerevisiae*. El conjunto de datos[19] que vamos a usar proviene del estudio realizado por Spellman y col. en [319] y también incluye datos de Cho y col [77].

Spellman y col. estudiaron el ciclo celular[20] de la levadura mediante datos de expresión genética provenientes de microarrays de ADN (la fotografía de uno de ellos se puede apreciar en la figura 7.1). El objetivo de este estudio era identificar aquellos genes que se expresaban durante el ciclo celular. Los experimentos fueron realizados con microarrays de aproximadamente 6200 genes, cuyas muestras fueron tomadas en 77 instantes bajo cuatro condiciones experimentales distintas, buscando capturar la expresión genética en las

[19] Disponible en http://cellcycle-www.stanford.edu/

[20] El ciclo celular es un conjunto ordenado de eventos que conducen al crecimiento de la célula y la división en dos células hijas.

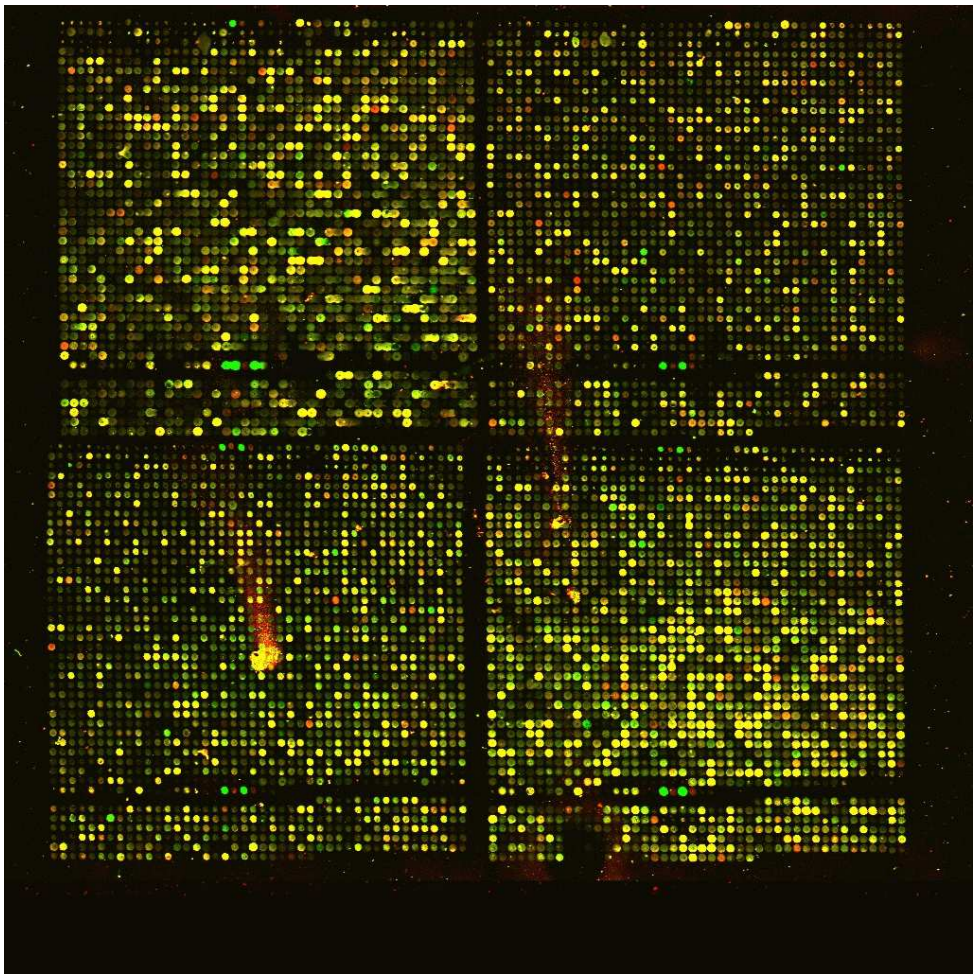

Figura 7.1: Imagen de uno de los microarrays de ADN usado por Spellman y colaboradores en sus experimentos. (Fuente http://cellcycle-www.stanford.edu/).

diferentes etapas del ciclo celular. El estudio de Spellman y col. identificó 800 genes expresados durante el ciclo celular.

De los 800 genes descubiertos, sólo pudieron explicar en su estudio el mecanismo de expresión para aproximadamente la mitad de ellos. Las bases en la regulación de los genes restantes, así como su función exacta en el ciclo celular, les era desconocida.

Los mecanismos de regulación genética del ciclo celular se conservan desde organismos más simples, como la levadura, hasta seres más complejos, como los mamíferos. Por tanto, las conclusiones obtenidas del estudio del hongo unicelular *Saccharomyces cerevisiae* pueden ser útiles incluso en el estudio de terapias contra el cáncer, ya que se conoce que muchos tumores son el resultado de una multitud de pasos, de los que una mutación no reparada del ADN en la fase de división celular podría ser el primer paso. Las alteraciones resultantes hacen que las células mutadas inicien un proceso de proliferación descontrolada e invadan tejidos normales. El desarrollo de un tumor maligno requiere de muchas transformaciones genéticas.

Las células que no están en división no se consideran que estén en el ciclo celular. Las etapas del ciclo celular, mostradas en la figura 7.2, son G1, S, G2 y M:

- El estado G1 quiere decir crecimiento (del inglés *growth*) o *GAP 1* (intervalo 1). Es la primera fase del ciclo celular, y en este estado la célula duplica su tamaño y masa.
- El estado S quiere decir *síntesis*. En esta fase ocurre la replicación o síntesis del ADN, por tanto, cada cromosoma se replica y queda formado por dos cromátidas idénticas.
- El estado G2 quiere decir *GAP 2* (intervalo 2). Es otra etapa de crecimiento de la célula donde, al final, se observan cambios en la estructura, que indican el principio de la división celular.
- El estado M agrupa a la mitosis (reparto de material genético nuclear) y a la citocinesis (división del citoplasma). Es la división celular en la que una célula progenitora se divide en dos nuevas células hijas idénticas.

### 7.2.2. Conocimiento experto.

Ante la imposibilidad de acceder a un experto en la materia, la solución adoptada para obtener conocimiento experto pasa por basarnos en trabajos de investigación aportados por otros autores. Concretamente, nos hemos basado en la red del ciclo celular de la levadura *Saccharomyces cerevisiae*

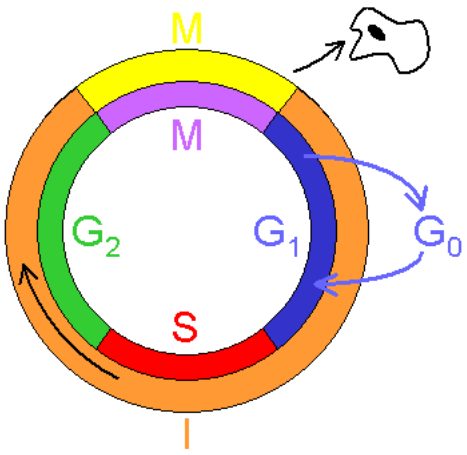


Figura 7.2: Etapas del ciclo celular. (Fuente Wikimedia http://commons.wikimedia.org/wiki/File:Cell_cycle.png).

obtenida por Nariai y col. en [243] y que se puede observar en la figura 7.3. Para construir dicha red utilizaron datos del trabajo de Spellman y col. [319] y del trabajo de Hughes y col. [159] . También obtuvieron información de la base de datos MIPS[21] [232].

Para obtener cada tipo de restricciones nos hemos basado en esta red, que denominaremos red original. En concreto, si $G = (\mathbf{X}, E_G)$ es la red original, entonces cada arco $X \rightarrow Y \in E_G$ es una posible restricción de existencia (se puede seleccionar la restricción $X \rightarrow Y \in E_e$ si el arco está también presente en la representación del PDAG completo de $G$; en otro caso podemos usar la restricción $X — Y \in E_e$); cada arco $X \rightarrow Y \notin E_G$ es una posible restricción de ausencia (en caso de que también $Y \rightarrow X \notin E_G$ seleccionamos de manera aleatoria si usar la restricción $X \rightarrow Y \in E_a$ o $X — Y \in E_a$); finalmente, si hay un camino dirigido de $X$ a $Y$ en el PDAG completo de $G$ entonces $X \rightarrow Y \in E_o$ es una posible restricción de orden.

[21] Base de datos de genomas y secuencias de proteínas del centro de información de Munich para secuencias de proteínas. Disponible en http://mips.gsf.de/

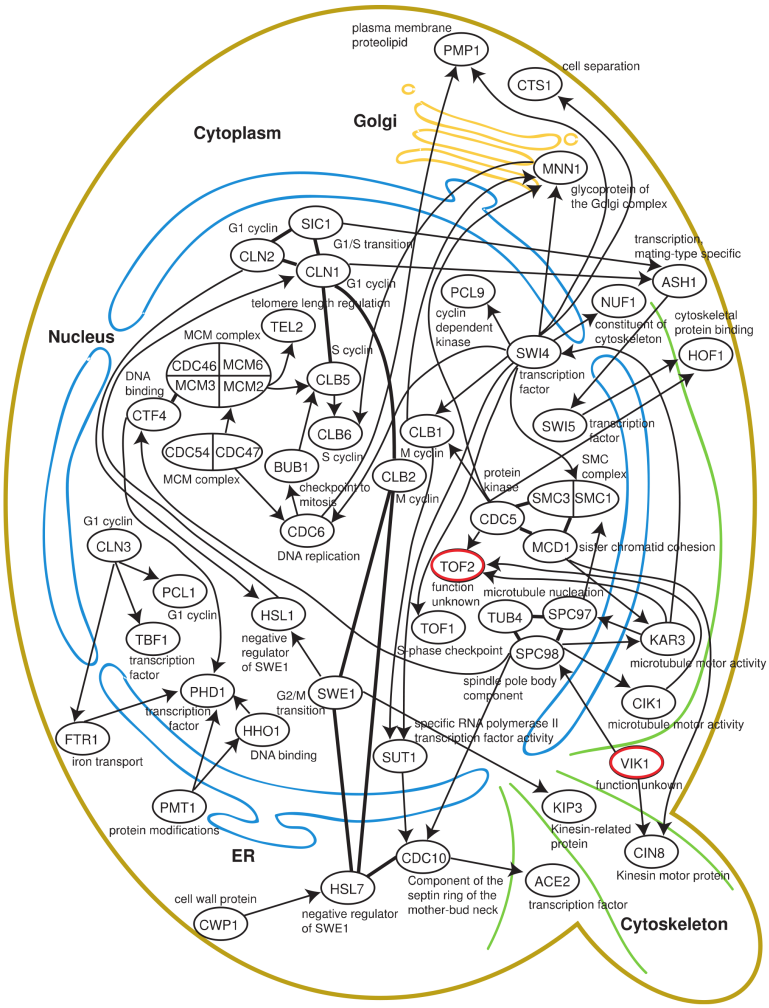


Figura 7.3: Red genética del ciclo celular de la levadura *Saccharomyces Cerevisiae*, obtenida por Nariai y colaboradores [243].

### 7.2.3. Preprocesamiento de los datos.

Para poder trabajar con la base de datos de expresión genética del ciclo celular de la levadura proporcionada por Spellman y colaboradores, ha sido

necesario realizar primero una etapa de procesamiento de los datos.

En primer lugar, se han eliminado los datos perdidos o desconocidos mediante el algoritmo de imputación de datos por los mínimos cuadráticos [185] *LLSImpute*. Al igual, que en el procesamiento de los datos del apartado de clasificación, la implementación usada de dicho algoritmo es la que se encuentra en el paquete *pcaMethods* para el lenguaje estadístico *R*.

Los niveles de expresión se mueven en un rango de valores continuos. Por tanto es necesario discretizarlos de forma que los métodos propuestos puedan trabajar con este tipo de datos. Para ello, hemos optado por la opción que siguen la mayoría de los autores en la literatura: la discretización en tres estados (infraexpresados, normal y sobreexpresados) propuesta por Friedman y col. en [122] y que ya hemos descrito en este capítulo. Esta fase de discretización ha sido realizada con el entorno *WEKA*, usando una implementación propia.

Puesto que nuestro objetivo no es el estudio de funciones desconocidas de los genes involucrados en el ciclo celular, sino que es estudiar el uso de restricciones estructurales en el análisis de datos de expresión genética, hemos decidido quedarnos solamente con aquellos genes que aparezcan en la red genética de Nariai y colaboradores. Por tanto, el conjunto de variables que se van a estudiar se ha visto reducido a 54 genes.

### 7.2.4. Resultados experimentales.

En este apartado vamos a describir los experimentos que se han llevado a cabo para probar la utilización de restricciones en datos de expresión genética, y los resultados obtenidos. Para ello, se han utilizado las restricciones con dos algoritmos, uno basado en el enfoque métrica+búsqueda y otro basado en tests de independencia. Todo los resultados de esta sección se han obtenido usando la herramienta *Elvira*.

Como base de datos hemos utilizado el conjunto de datos de expresión genética de Spellman y col., con el preprocesamiento antes mencionado. Como red original nos hemos basado en la red propuesta por Nariai y col. (ver figura 7.3), pero orientando los enlaces que en la red original no estaba orien-

tados. Esta orientación de enlaces se ha realizado sin formar nuevos ciclos y evitando, en la medida de lo posible, formar nuevos patrones cabeza-cabeza. Como resultado obtenemos una red con 55 nodos y 100 enlaces.

Una vez obtenida la estructura de la red original, hemos realizado un aprendizaje paramétrico de la misma usando la base de datos de Spellman y colaboradores. En dicho aprendizaje se ha usado un estimador suavizado basado en la *Ley de la sucesión de Laplace* [137], con el fin de evitar los problemas de sobreajuste y falta de fiabilidad, que un estimador por máxima verosimilitud nos puede generar para este conjunto de datos con tan pocas muestras.

El algoritmo de aprendizaje basado en el enfoque métrica+búsqueda considerado es la búsqueda local clásica (con operadores de inserción, eliminación e inversión de arcos), usando la métrica bayesiana BDeu [150], donde el parámetro que representa el tamaño muestral equivalente es igual a 1 y se utiliza una estructura uniforme *a priori*. El algoritmo basado en tests de independencia es el PC. En ambos casos el aprendizaje paramétrico realizado ha sido el mismo que el efectuado en la red original.

Las medidas usadas para estudiar el comportamiento de los métodos son: (1) El valor de la métrica (BDeu) para la red obtenida; esta medida es interesante debido a que es el criterio que guía la búsqueda local. (2) Tres medidas de diferencia estructural entre la red aprendida y la red original, que miden la capacidad de reconstruir la estructura del grafo: número de arcos añadidos (A), número de arcos eliminados (D) y el número de arcos invertidos (I) en la red aprendida con respecto a la red original. Para descartar diferencias o semejanzas ficticias entre dos redes, causadas por diferentes pero equivalentes estructuras subDAG, antes de comparar, las dos redes se convierten a su correspondiente representación PDAG completo, usando el algoritmo propuesto en [72]. (3) Una medida de la habilidad para reconstruir la distribución de probabilidad conjunta: vamos a usar la divergencia de Kullback-Leibler (KL) entre las distribuciones de probabilidad asociadas a la red original y a la red aprendida.

Para los experimentos se han seleccionado de forma aleatoria unos porcentajes de restricciones de cada tipo, extraídas del conjunto de todas las posibles restricciones correspondientes a la red original. Los porcentajes de restricciones usados han sido 10 %, 20 %, 30 % y 40 %.

Se han ejecutado los algoritmos de aprendizaje para cada porcentaje de restricciones de cada tipo por separado, y también usando los tres tipos de restricciones a la vez. Los resultados que se muestran a continuación, representan los valores medios de las distintas medidas tomadas a lo largo de 50 iteraciones, es decir, se ha generado para cada porcentaje de restricciones 50 conjuntos de restricciones aleatorias y se ha ejecutado cada método de aprendizaje para cada uno de ellos.

La tabla 7.7 muestra los resultados obtenidos usando el algoritmo de búsqueda local, mientras que la tabla 7.9 muestra los resultados para el algoritmo PC. Incluyen los resultados obtenidos por el algoritmo de aprendizaje cuando no se usan restricciones (0 %). También muestran los valores de la divergencia KL de las redes originales, con los parámetros reaprendidos, lo cual nos sirve para comparar con las redes que sí usan restricciones. La tabla 7.8 muestra los valores medios de la métrica BDeu en las distintas redes aprendidas con búsqueda local, así como el valor de la métrica de la red original.

| | $G_e$, $G_a$, $G_o$ | | | | sólo $G_e$ | | | | sólo $G_a$ | | | | sólo $G_o$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| % | KL | A | D | I | KL | A | D | I | KL | A | D | I | KL | A | D | I |
| 10 % | 12,6290 | 51,9 | 76,0 | 4,3 | 13,1117 | 56,0 | 77,1 | 4,8 | 13,1841 | 54,8 | 85,4 | 5,1 | 13,7988 | 57,7 | 85,4 | 4,8 |
| 20 % | 11,7050 | 45,9 | 67,3 | 3,2 | 12,4809 | 52,1 | 68,6 | 4,2 | 12,5127 | 51,2 | 84,0 | 5,7 | 13,6561 | 58,9 | 85,0 | 4,6 |
| 30 % | 10,7688 | 40,0 | 58,8 | 1,8 | 12,1238 | 50,8 | 60,1 | 3,5 | 11,7540 | 47,8 | 82,7 | 5,6 | 13,4880 | 58,2 | 84,5 | 4,2 |
| 40 % | 9,7982 | 34,9 | 50,0 | 1,3 | 11,7335 | 49,1 | 51,2 | 2,9 | 10,8754 | 43,2 | 81,4 | 5,2 | 13,5215 | 58,7 | 84,4 | 3,8 |
| 0 % | 13,6558 | 57,0 | 87,0 | 4,0 | KL red original (reaprendida): 7,7579 | | | | | | | | | | | |

Tabla 7.7: Resultados medios obtenidos usando búsqueda local.

Primero analicemos los resultados de la búsqueda local desde el punto de vista de las diferencias estructurales. Se observa que el número de arcos eliminados, añadidos e invertidos decrece a la par que el número de restricciones de existencia, ausencia y orden, respectivamente, crece. Además, se observa que, en general, el uso de cualquiera de los tres tipos de restricciones tiende también a disminuir las otras medidas de diferencia estructural. Por ejemplo, la restricciones de existencia disminuyen el número de arcos borrados, pero también el número de arcos añadidos o invertidos. Como excepciones a esta última observación, no vemos que el número de arcos invertidos decrezca claramente con el uso de restricciones de ausencia o que el número de arcos añadidos disminuya con el uso de restricciones de orden.

Con respecto al análisis de los resultados desde el punto de vista de la divergencia KL, el uso de cada tipo de restricción conduce a mejores estructuras de red. Las mejoras se incrementan sistemáticamente con el número de restricciones usadas. Como se puede esperar, dicha mejoría es mayor cuando utilizamos los tres tipos de restricciones a la vez.

| | $G_e$, $G_a$, $G_o$ | sólo $G_e$ | sólo $G_a$ | sólo $G_o$ |
|---|---|---|---|---|
| 10 % | -02509,46 | -02474,52 | -02440,86 | -02432,29 |
| 20 % | -02596,95 | -02530,98 | -02469,69 | -02442,33 |
| 30 % | -02689,36 | -02587,79 | -02504,13 | -02448,65 |
| 40 % | -02783,47 | -02647,72 | -02542,60 | -02452,42 |
| 0 % | -2418,38 | BDeu red original:-3420,68 | | |

Tabla 7.8: Valores medios de la métrica BDeu usando búsqueda local.

En lo que respecta a los valores de la métrica BDeu, se puede observar que las redes aprendidas siempre tienen valores menores de la métrica BDeu que la red original. No obstante, conforme se incrementa el número de restricciones, los valores de la métrica BDeu tienden a incrementarse, acercándose progresivamente a la red original. El uso de ningún tipo de restricción en el algoritmo de aprendizaje muestra el peor resultado para la métrica.

| | $G_e$, $G_a$, $G_o$ | | | | sólo $G_e$ | | | | sólo $G_a$ | | | | sólo $G_o$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| % | KL | A | D | I | KL | A | D | I | KL | A | D | I | KL | A | D | I |
| 10 % | 8,0912 | 5,4 | 85,7 | 0,5 | 8,1300 | 6,0 | 86,2 | 0,8 | 8,6618 | 5,4 | 96,5 | 0,0 | 8,7287 | 5,9 | 96,9 | 0,0 |
| 20 % | 7,5065 | 4,9 | 75,4 | 0,7 | 7,6013 | 6,0 | 76,4 | 1,2 | 8,5528 | 4,9 | 96,0 | 0,0 | 8,7236 | 5,6 | 96,6 | 0,0 |
| 30 % | 7,0433 | 5,4 | 65,1 | 1,0 | 7,1439 | 6,0 | 66,5 | 1,4 | 8,4120 | 4,8 | 92,2 | 0,0 | 8,7180 | 5,3 | 96,3 | 0,0 |
| 40 % | 6,7435 | 6,8 | 54,8 | 0,9 | 6,7672 | 5,9 | 56,7 | 1,4 | 8,3110 | 5,6 | 94,2 | 0,0 | 8,7156 | 5,2 | 96,2 | 0,0 |
| 0 % | 8,7392 | 6,0 | 97,0 | 0,0 | KL red original (reaprendida): 7,7579 | | | | | | | | | | | |

Tabla 7.9: Resultados medios obtenidos usando PC.

Como ocurría en la búsqueda local, en el caso del algoritmo PC, y desde el punto de vista de la divergencia KL, el uso de restricciones siempre conduce a mejores estructuras de red que el algoritmo PC sin restricciones.

Con respecto a las diferencias estructurales, el uso de restricciones muestra una reducción de las diferencias estructurales con la red original. También

observamos que el uso de cualquiera de los tres tipos de restricciones tiende también a disminuir las otras medidas de diferencia estructural. Como excepción, a esta última observación, el número de arcos invertidos crece levemente con el uso de restricciones de existencia.

Tenemos que hacer mención a los pobres resultados obtenidos por los distintos algoritmos de aprendizaje en su obtención de la red original. Nótese que de los 100 enlaces que posee, el algoritmo PC sin restricciones elimina 97 y el algoritmo de búsqueda local, también sin restricciones, elimina 87. Respecto a los arcos que no están en la red original y que sí encuentran ambos algoritmos sin restricciones, podemos observar que el algoritmo de búsqueda local añade 57 arcos y el algoritmo basado en tests de independencias añade 6 enlaces. La explicación a estos pobres resultados en los algoritmos de aprendizaje los encontramos en el número tan escaso de muestras (sólo 77 casos) con los que tienen que construir la estructura de la red. En efecto, si nos concentramos en el algoritmo PC, observamos que sólo aprende 9 enlaces (6 añadidos y 3 que estaban a la red original) y esto sucede debido a que los tests de independencia, al trabajar con tan pocas muestras, toman como independientes la gran mayoría de los tests que realiza.

No obstante, y a pesar de un punto de partida tan malo, se ve claramente que el uso de restricciones mejora siempre los resultados que se obtienen para todas las medidas tomadas. Cuando trabajamos con tan pocas muestras, y como era de esperar, esto nos indica que, primero, el uso de conocimiento experto en los algoritmos de aprendizaje va a redundar en una mejora de los resultados y, segundo, cuanto mayor es el conocimiento aportado mejores resultados se obtienen.

## 7.3. Discusión.

En este capítulo se ha estudiado la aplicación al análisis de datos de expresión genética, de las distintas propuestas aportadas en esta memoria. Estos datos tienen como principal inconveniente la maldición de la dimensionalidad, esto es, tenemos muchas variables (genes) para estudiar pero con muy pocas muestras para hacerlo.

En el apartado de los clasificadores bayesianos, podemos ver que los re-

sultados obtenidos son mejores que los encontrados en la literatura para los distintos problemas estudiados, a excepción de los árboles de clasificación usando una estimación bayesiana. No obstante, observamos un peor comportamiento en las multiredes bayesianas, aún cuando usamos una búsqueda por envoltura. La explicación de este hecho creemos que está en la partición del conjunto de datos para construir las distintas subredes, hipótesis que creemos confirmada con los malos resultados obtenidos por los árboles de clasificación. El hecho de que las multiredes bayesianas puedan modelizar independencias dependientes del contexto (presentes en los contextos celulares) queda anulado al trabajar con un conjunto de datos que posee tan pocas muestras (y que disminuye aún más al hacer la correspondiente partición para cada red de la multired).

Por otro lado, observamos los excelentes resultados que muestra el clasificador C-RPDAG debido a dos motivos: uno, tiene una mayor capacidad expresiva que los clasificadores bayesianos clásicos (naïve bayes o TAN) y que están limitados estructuralmente; dos, usa un espacio de búsqueda especializado que nos permite centrar la búsqueda desde la perspectiva de la clasificación y explorar un espacio de búsqueda más robusto y reducido.

Desde el punto de vista del uso de restricciones estructurales en el aprendizaje de redes bayesianas para el análisis de datos de expresión genética, obtenemos que, aún cuando el punto de partida sea relativamente malo, el uso de conocimiento experto en forma de restricciones mejora siempre los resultados obtenidos. Esta forma gráfica de incluir conocimiento la consideramos bastante apropiada e intuitiva, a la hora de usar información sobre relaciones entre genes y, por tanto, útil en el análisis de datos de expresión genética.

# Capítulo 8

# Conclusiones y Principales Aportaciones

Si bien, en los distintos capítulos hemos visto las conclusiones en la discusión del mismo, en este último capítulo se presentan las conclusiones generales de esta memoria y principales aportaciones. Además también presentaremos una lista de publicaciones y líneas de investigación futuras continuando con el trabajo aquí empezado.

Todo este trabajo ha girado en torno a la aplicación de nuevos modelos bayesianos al análisis de datos de expresión genética. Se han presentado nuevos clasificadores bayesianos que muestran un buen comportamiento tanto en problemas clásicos de aprendizaje automático como en problemas de datos provenientes de microarrays de ADN. También se ha presentado una metodología para la incorporación de conocimiento experto en el aprendizaje de redes bayesianas, evaluando dicha metodología con distintos problemas clásicos, así como con datos de expresión genética.

Al principio de esta memoria, hemos visto que las redes bayesianas nos permiten representar el conocimiento de forma gráfica y compacta, usando los conceptos de probabilidad y causalidad o independencia entre las variables de un problema. Se han presentado distintos métodos de aprendizaje de redes bayesianas existentes en la literatura y su uso en tareas de clasificación supervisada, destacando los clasificadores bayesianos.

Seguidamente se ha realizado un exhaustivo análisis de la literatura exis-

tente en la aplicación de las redes bayesianas a datos de expresión genética. Observando que se consideran la mejor herramienta que hay en la literatura para la inferencia en redes de regulación genética. Esto se debe a una serie de ventajas de las redes bayesianas. Por ejemplo, poseen sólidas bases estadísticas, determinan relaciones estocásticas entre genes, pueden describir procesos locales, proporcionan modelos de influencia causal, permiten la incorporación de conocimiento experto, se pueden usar en clasificación, son modelos visualmente interpretables y utilizan la probabilidad como medida de incertidumbre por lo que son indicadas para trabajar con datos con gran cantidad de ruido. No obstante, las redes bayesianas se tienen que enfrentar al problema de la dimensionalidad que poseen estos datos: se estudian muchos genes pero hay pocas muestras para hacerlo.

Posteriormente, se han definido formalmente tres tipos de restricciones para el aprendizaje estructural de redes bayesianas, que hemos denominado restricciones de existencia, ausencia y orden. Las restricciones presentadas nos permiten incorporar conocimiento de forma visualmente interpretable, siendo así una herramienta intuitiva en su utilización por un experto. Hemos estudiado su uso en combinación con algoritmos de aprendizaje, basados en métrica+búsqueda y en tests de independencia, concretamente, hemos ilustrado su uso para la búsqueda local y el algoritmo PC. Los resultados experimentales muestran que el uso de información adicional en forma de restricciones mejora las estructuras de las redes obtenidas.

Se ha presentado un nuevo método para la construcción de árboles de clasificación y se ha comparado con otros métodos clásicos. El método desarrollado es una variante de estos algoritmos clásicos, pues se ha usado una distribución *a priori* de Dirichlet para el cálculo de la probabilidad. Hemos visto que con esta modificación, se han mejorado los resultados en precisión y en tiempo.

Las multiredes bayesianas son una extensión de las redes bayesianas en tanto en cuanto nos permiten representar independencias condicionales dependientes del contexto. Se ha realizado un estudio de diferentes métodos para obtener la variable distinguida en las multiredes bayesianas, cuando ésta es un atributo. Después de probar numerosas alternativas se han destacado tres métodos de obtención de la variable distinguida en función de las necesidades del problema con el que estemos trabajando.

Es común la creencia que un enfoque métrica+búsqueda puede construir redes bayesianas que maximizan el valor de la métrica pero que se comportan pobremente como clasificadores. Esto es una verdad a medias, debido a que el uso de un algoritmo de búsqueda especializado en clasificación nos permite obtener mejores resultados que otros algoritmos de búsqueda no orientados a clasificación, aún usando un enfoque métrica+búsqueda. De hecho, se ha desarrollado un nuevo método basado en el paradigma métrica+búsqueda para aprender clasificadores bayesianos sin restricciones estructurales. Este nuevo método está basado en el uso de C-RPDAGs, un tipo de grafos acíclicos parcialmente dirigidos, como elementos del espacio de búsqueda. Combinan las ideas de equivalencia en clasificación y equivalencia en independencia, para producir un espacio de búsqueda más reducido, focalizado y robusto. Los resultados experimentales muestran que el método propuesto obtiene, de manera eficiente, clasificadores que compiten favorablemente con otros clasificadores bayesianos del estado del arte.

Finalmente, se ha realizado un estudio de la aplicación de las distintas propuestas aportadas en esta memoria al análisis de datos de expresión genética. En el apartado de los clasificadores bayesianos, hemos obtenido mejores resultados que los encontrados en la literatura para los distintos problemas estudiados. Desde el punto de vista del uso de restricciones estructurales en el aprendizaje de redes bayesianas para el análisis de datos de expresión genética, obtenemos que, aún cuando el punto de partida sea relativamente malo, el uso de conocimiento experto en forma de restricciones mejora siempre los resultados obtenidos.

## Trabajos publicados.

Relacionados con el desarrollo de esta memoria se han publicado los siguientes trabajos como autor principal:

- L. de Campos, J. G. Castellano. "Bayesian network learning algorithms using structural restrictions". *International Journal of Approximate Reasoning*. Volumen 45, Issue 2, pp 233-254, 2007.
- A. Cano, J. G. Castellano, A. R. Masegosa , S. Moral. "Uso de redes

bayesianas en el análisis de datos de microarrays de ADN". *VI Jornadas de Transferencia Tecnológica de Inteligencia Artificial, TTIA'2005 (AEPIA). I Congreso Español De Informática (CEDI 2005)*, ISBN: 84-9732-435-8, pp. 89-98, Granada, España. 2005.

- A. Cano, J. G. Castellano, A. R. Masegosa , S. Moral. "Methods to determine the branching attribute in Bayesian multinets classifiers". En *Eight European Conferences on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2005)*, L. Godo (Ed.), LNAI 3571, ISBN 3-540-27326-3, pp. 932-943, Barcelona, España. 2005.

- L. de Campos, J. G. Castellano. "On the use of restrictions for learning Bayesian networks". En *Eight European Conferences on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2005)*, L. Godo (Ed.), LNAI 3571, ISBN 3-540-27326-3, pp. 174-185, Barcelona, España. 2005.

- S. Acid, L. de Campos, J. G. Castellano. "Learning Bayesian network classifiers: Searching in a space of partially directed acyclic graphs". *Machine Learning*. ISSN: 0885-6125. Volumen 59, número 3. pp. 213-235. 2005.

- A. Cano, J. G. Castellano, S. Moral."Árboles de clasificación usando una estimación bayesiana". En *X conferencia de la Asociación Española para la Inteligencia Artificial CAEPIA-TTIA'2003*. pp. 97-100 , ISBN:84-8373-564-4, Donostia-San Sebastián (España). 2003.

Relacionados con el tema tratado en esta memoria también se ha colaborado en las siguientes publicaciones:

- A. Cano, J. G. Castellano, A. R. Masegosa , S. Moral. "Aplicación de un modelo naïve bayes gaussiano con selección de variables al análisis de datos de expresión genética". En *VI Jornadas de Transferencia Tecnológica de Inteligencia Artificial, TTIA'2005 (AEPIA). I Congreso Español De Informática (CEDI 2005)*, ISBN: 84-9732-435-8, pp 81-88, Granada, España. 2005.

- A. Cano, J. G. Castellano, A. R. Masegosa , S. Moral. "Selective Gaussian naïve Bayes model for diffuse large-B-cell lymphoma classification". En *Eight European Conferences on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2005)*, L.

Godo (Ed.), LNAI 3571, ISBN 3-540-27326-3, pp. 908-920, Barcelona, España. 2005.

- A. Cano, J. G. Castellano, A. R. Masegosa , S. Moral. "Application of a selective Gaussian naïve Bayes model for diffuse large-B-cell lymphoma classification". En *Proceedings of the 2nd European Workshop on Probabilistic Graphical Models(PGM'04)*, pp 33-40. Leiden, The Netherlands. 2004.

- Elvira Consortium. "Elvira: an environment for Probabilistic Graphical Models". En *Procs. Of the 1st European Workshop on Probabilistic Graphical Models. (PGM 2002)*, J. A. Gámez an A. Salmerón (Eds), pp 222-230. Cuenca (España). 2002.

## Lineas de investigación futuras.

Los continuos avances que están experimentando tecnologías como los microarrays de ADN, permitirán en un futuro cercano obtener una mayor cantidad de datos de expresión genética, paliando de esta forma, y en cierta medida, la maldición de la dimensionalidad de estos datos. No obstante, hay multitud de nuevas técnicas que están apareciendo en biología generando gran cantidad de datos, donde es necesario extraer información útil y donde las distintas técnicas de aprendizaje de datos van a ser claves para el análisis y conocimiento de este tipo de información. Por tanto, hay una gran diversidad de nuevas posibilidades en las que poder continuar nuestra labor de investigación.

Siendo más específicos y pensando en las nuevas propuestas que se han hecho en esta memoria, queremos estudiar distintas mejoras y seguir varias líneas de trabajo en un futuro inmediato que nos parecen interesantes.

En el uso de restricciones estructurales nos parece relevante estudiar el comportamiento del enfoque propuesto con restricciones blandas, especialmente en el análisis de datos de expresión genética, pues hay varios autores que utilizan esta segunda vía. También estamos interesados en aplicar los métodos propuestos a datos provenientes de microarrays de ADN, pero no sólo para contrastar su validez como se ha hecho en esta memoria, sino con

el objetivo de extraer información biológica que sea relevante.

En el caso de las multiredes bayesianas, queremos estudiar aproximaciones híbridas de filtrado y envoltura que obtengan un buen rendimiento en tanto por ciento de bien clasificados en un menor tiempo. Estas técnicas consistirían en hacer un ranking con el método de filtrado y de ese ordenamiento obtener un subconjunto de variables en la que se haría la búsqueda por envoltura. Además, nos parece interesante estudiar el comportamiento de las multiredes bayesianas recursivas basándonos en el conocimiento obtenido con los árboles de clasificación usando una estimación bayesiana. Otra idea en la que trabajar, es utilizar clasificadores bayesianos más potentes en las hojas que el clasificador naïve bayes. También estamos interesados en modelizar datos dinámicos usando multiredes bayesianas donde la variable distinguida es una variable temporal.

Para el clasificador C-RPDAG, estamos especialmente interesados en buscar una métrica apropiada. Puesto que nos hemos basado en un enfoque métrica+búsqueda y hemos especializado el algoritmo de búsqueda en tareas de clasificación, ahora nos parece relevante hacer un estudio sobre diferentes métricas, como por ejemplo, usar la métrica MIT, probar enfoques de envoltura, usar la métrica BDeu donde el parámetro que representa el tamaño muestral equivalente sea distinto a 1, etc. También sería interesante probar algoritmos de búsqueda más avanzados que la búsqueda local. Por otro lado, nos parece muy conveniente, mejorar la eficiencia de este clasificador si queremos seguir utilizándolo en datos con un gran número de variables, como ocurre en los datos de expresión genética o en los conjuntos de datos que siguen proporcionando las distintas ramas de investigación en biología.

# Referencias Bibliográficas


[1] J. Abellán y A. R. Masegosa. Split criterions for variable selection using decision trees. En "ECSQARU '07: Proceedings of the 9th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty", págs. 489–500. Springer-Verlag (2007).

[2] J. Abellán y S. Moral. Building classification trees using the total uncertainty criterion. En "ISIPTA'01, Proceedings of the Second Symposium on Imprecise Probabilities and Their Applications", págs. 1–8. Shaker Publishing (2001).

[3] B. Abramson, J. Brown, W. Edwards, A. Murphy y R. L. Winkler. Hailfinder: A Bayesian system for forecasting severe weather. *International Journal of Forecasting* **12**(1), 57–71 (1996).

[4] S. Acid y L. M. de Campos. Learning right sized belief networks by means of a hybrid methodology. En D. A. Zighed, H. J. Komorowski y J. M. Zytkow, editores, "Principles of Data Mining and Knowledge Discovery, 4th European Conference, PKDD 2000", vol. 1910 de "Lecture Notes in Computer Science", págs. 309–315. Springer (2000).

[5] S. Acid y L. M. de Campos. A hybrid methodology for learning belief networks: BENEDICT. *International Journal of Approximate Reasoning* **27**(3), 235–262 (2001).

[6] S. Acid y L. M. de Campos. Searching for Bayesian network structures in the space of restricted acyclic partially directed graphs. *Journal of Artificial Intelligence Research* **18**, 445–490 (2003).

[7] H. Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on* **19**(6), 716–723 (1974).

[8] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts y J. D. Watson. "Biología molecular de la célula". Omega S.A., España, 3 ed. (2002).

[9] A. Alizadeh, M. Eisen, E. Davis, C. Ma, I. Lossos, A. Rosenwald, J. Boldrick, H. Sabet, T. Tran, X. Yu, J. Powell, L. Yang, G. Marti, J. H. Jr, L. Lu, D. Lewis, R. Tibshirani, G. Sherlock, W. Chan, T. Greiner, D. Weisenburger, J. Armitage, R. Warnke, R. Levy, W. Wilson, M. Grever, J. Byrd, D. Botstein, P. Brown y L. Staudt. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature* **403**, 503–511 (2000).

[10] P. D. Allison. "Missing data". Quantitative Applications in the Social Sciences. Thousand Oaks (2002).

[11] E. Almasri, P. Larsen, G. Chen y Y. Dai. Incorporating literature knowledge in Bayesian network for inferring gene networks with gene expression data. En I. I. Mandoiu, R. Sunderraman y A. Zelikovsky, editores, "Bioinformatics Research and Applications, Fourth International Symposium, ISBRA 2008. Proceedings", vol. 4983, págs. 184–195 (2008).

[12] U. Alon, N. Barkai, D. A. Notterman, K. Gishdagger, S. Ybarradagger, D. Mackdagger y A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America* **96**(12), 6745–6750 (1999).

[13] S. Andersson, D. Madigan y M. D. Perlman. A characterization of Markov equivalence classes for acyclic digraphs. *Annals of Statistics* **25**, 505–541 (1997).

[14] R. Armañanzas, B. Calvo, I. Inza, P. Larrañaga, I. Bernales, A. Fullaondo y A. M. Zubiaga. Bayesian classifiers with consensus gene selection: A case study in the systemic lupus erythematosus. En L. L. Bonilla, M. Moscoso, G. Platero y J. M. Vega, editores, "Progress in Industrial Mathematics at ECMI 2006", vol. 12 de "Mathematics in Industry", págs. 560–565. Springer (2007).

[15] R. Armañanzas. Medidas de filtrado de selección de variables mediante la plataforma Elvira. Proyecto Fin de Carrera, Computer Science and Artificial Intelligence department, University of the Basque Country (2004).

[16] R. Armañanzas. Solving bioinformatics problems by means of Bayesian classifiers and feature selection. Technical Report EHU-KZAA-IK-2/06, University of the Basque Country (2006).

[17] R. Armañanzas, I. Inza y P. Larrañaga. Detecting reliable gene interactions by a hierarchy of Bayesian network classifiers. *Computer Methods and Programs in Biomedicine* **91**(2), 110–121 (2008).

[18] S. A. Armstrong, J. E. Staunton, L. B. Silverman, R. Pieters, M. L. den Boer, M. D. Minden, S. E. Sallan, E. S. Lander, T. R. Golub y S. J. Korsmeyer. MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nature Genetics* **30**, 41 – 7 (2002).

[19] P. Baldi y G. W. Hatfield. "DNA microarrays and gene expression. From experiments to data analysis and modelling." Cambridge University Press (2002).

[20] P. Baldi y A. D. Long. A Bayesian framework for the analysis of microarray expression data: Regularized t-test and statistical inferences of gene changes. *Bioinformatics* **17**, 509–519 (2001).

[21] H. Bannai, S. Inenaga, A. Shinohara, M. Takeda y S. Miyano. A string pattern regression algorithm and its application to pattern discovery in long introns. *International Conference on Genome Informatics* **13**, 3–11 (2002).

[22] H. Bannai, S. Inenaga, A. Shinohara, M. Takeda y S. Miyano. Efficiently finding regulatory elements using correlation with gene expression. *Journal of Bioinformatics and Computational Biology* **2**(2), 273–88 (2004).

[23] M. Bansal, V. Belcastro, A. Ambesi-Impiombato y D. di Bernardo. How to infer gene networks from expression profiles. *Molecular Systems Biology* **3** (2007).

[24] Z. Bar-Joseph. Analyzing time series gene expression data. *Bioinformatics* **20**(16), 2493–503 (2004).

[25] A. Barabási y Z. Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics* **5**, 101–113 (2004).

[26] Y. Barash y N. Friedman. Context-specific Bayesian clustering for gene expression data. *Journal of Computational Biology* **9**(2), 169–91 (2002).

[27] G. Bastos y K. S. Guimarães. Analyzing the effect of prior knowledge in genetic regulatory network inference. En S. K. Pal, S. Bandyopadhyay y S. Biswas, editores, "Pattern Recognition and Machine Intelligence, First International Conference, PReMI 2005, Proceedings", vol. 3776 de "Lecture Notes in Computer Science", págs. 611–616. Springer (2005).

[28] G. Bastos y K. S. Guimarães. A simpler Bayesian network model for genetic regulatory network inference. En "Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on", vol. 1, págs. 304– 309 (2005).

[29] I. A. Beinlich, H. J. Suermondt, R. M. Chavez y G. F. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. En "Second European Conference on Artificial Intelligence in Medicine", vol. 38, págs. 247–256. Springer-Verlag (1989).

[30] R. E. Bellman. "Dynamic programming". Princeton University Press (1957).

[31] M. Ben-Bassat. Use of distance measures, information measures and error bounds in feature evaluation. En "Handbook of Statistics", vol. 2, págs. 773–791. North-Holland Publishing Company, Amsterdan (1982).

[32] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachmana, M. Schummer y Z. Yakhini. Tissue classification with gene expression profiles. *Journal of Computational Biology* **7**(3-4), 559–83 (2000).

[33] A. Ben-Dor, N. Friedman y Z. Yakhini. Scoring genes for relevance. Informe técnico 2000-38, School of Computer Science & Engineering, Hebrew University, Jerusalem (2000).

[34] A. Ben-Dor, N. Friedman y Z. Yakhini. Class discovery in gene expression data. En "RECOMB '01: Proceedings of the fifth annual international conference on Computational biology", págs. 31–38. ACM Press (2001).

[35] A. Bernard y A. J. Hartemink. Informative structure priors: joint learning of dynamic regulatory networks from multiple types of data. *Proceedings of pacific symposium on biocomputing* págs. 459–70 (2005).

[36] A. Berns. Cancer: Gene expression in diagnosis. *Nature* **403**, 491–492 (2000).

[37] J. C. Bezdek. "Pattern recognition and machine inteligence cognition with fuzzy objective function algorithms (Advanced applications in pattern recognition)". Springer (1981).

[38] A. Bhattacharjee, W. G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno, M. Gillette, M. Loda, G. Weber, E. J. Mark, E. S. Lander, W. Wong, B. E. Johnson, T. R. Golub, D. J. Sugarbaker y M. Meyerson. Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proceedings of the National Academy of Sciences of the United States of America* **98**, 13790 – 5 (2001).

[39] J. Binder, D. Koller, S. Russell, K. Kanazawa y P. Smyth. Adaptive probabilistic networks with hidden variables. *Machine Learning* **29**(2–3), 213–244 (1997).

[40] C. Blake y C. Merz. UCI repository of machine learning databases. `http://www.ics.uci.edu/~mlearn/MLRepository.html`. University of California, Irvine, Dept. of Information and Computer Sciences. (1998).

[41] W. J. Blake, M. Kærn, C. R. Cantor y J. J. Collins. Noise in eukaryotic gene expression. *Nature* **422**(6932), 633–637 (2003).

[42] R. Blanco. "Learning Bayesian Networks from Data with Factorisation and Classification Purposes. Applications in Biomedicine." Tesis Doctoral, University of the Basque Country, Spain (2005).

[43] R. Blanco, I. Inza y P. Larrañaga. Learning Bayesian networks in the space of structures by estimation of distribution algorithms. *International Journal of Intelligent Systems* **18**(2), 205–220 (2003).

[44] R. Blanco, P. Larrañaga, I. Inza y B. Sierra. Selection of highly accurate genes for cancer classification by estimation of distribution algorithms. En "Proceedings of the Workshop Bayesian Models in Medicine, held within AIME 2001, Artificial Intelligence in Medicine", págs. 29–34 (2001).

[45] F. R. Blattner, G. Plunkett, C. A. Bloch, N. T. Perna, V. Burland, M. Riley, J. Collado-Vides, J. D. Glasner, C. K. Rode, G. F. Mayhew, J. Gregor, N. W. Davis, H. A. Kirkpatrick, M. A. Goeden, D. J. Rose, B. Mau y Y. Shao. The complete genome sequence of Escherichia coli K-12. *Science* **277**(5331), 1453–74 (1997).

[46] D. M. Blei, A. Y. Ng y M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research* **3**, 993–1022 (2003).

[47] J. Bockhorst, M. Craven, D. Page, J. Shavlik y J. Glasner. A Bayesian network approach to operon prediction. *Bioinformatics* **19**(10), 1227–35 (2003).

[48] A. Bosin, N. Dessì, D. Liberati y B. Pes. Learning Bayesian classifiers from gene-expression microarray data. En I. Bloch, A. Petrosino y A. Tettamanzi, editores, "Fuzzy Logic and Applications, 6th International Workshop, WILF 2005, Revised Selected Papers", vol. 3849 de "Lecture Notes in Computer Science", págs. 297–304. Springer (2005).

[49] R. Bouckaert. Belief networks construction using the minimum description length principle. En M. Clarke, R. Kruse y S. Moral, editores, "Symbolic and Quantitative Approaches to Reasoning and Uncertainty, Lecture Notes in Computer Science", vol. 747, págs. 41–48. Springer (1993).

[50] R. Bouckaert. "Bayesian belief networks: from construction to inference". Tesis Doctoral, University Utrecht (1995).

[51] C. Boutilier, N. Friedman, M. Goldszmidt y D. Koller. Context-specific independence in Bayesian networks. *Uncertainty in Artificial Intelligence. Proceeedings of the Twelfth conference.* págs. 115–123 (1996).

[52] D. Bowser-Chao y L. Debra. A comparison of the use of binary decision trees and neural networks in top quark detection. *Physical Review D: Particles and Fields* **47**, 1900–1905 (1993).

[53] L. Breiman, J. H. Friedman, J. A. Olshen y C. J. Stone. "Classification and regression trees". Wadsworth & Brooks/Cole Advanced Books & Software (1984).

[54] P. Broët, S. Richardson y F. Radvanyi. Bayesian hierarchical model for identifying changes in gene expression from microarray experiments. *Journal of Computational Biology* **9**(4), 671–683 (2002).

[55] M. P. S. Brown, W. Grundy, D. Lin, N. Cristianini, C. Sugnet, T. S. Furey, J. M.Ares y D. Haussler. Knowledge-based analysis of microarray gene expression data using support vector machines. *Proceedings of the National Academy of Sciences of the United States of America (PNAS)* **97**(1), 262–267 (2000).

[56] W. L. Buntine. Theory refinement on Bayesian networks. En "Proceedings of the 7th Annual Conference on Uncertainty in Artificial Intelligence (UAI-91)", págs. 52–60. Morgan Kaufmann Publishers (1991).

[57] W. L. Buntine. Operations for learning with graphical models. *Journal of Artificial Intelligence Research* **2**, 159–225 (1994).

[58] W. L. Buntine. A guide to the literature on learning probabilistic networks from data. *IEEE Transactions on Knowledge and Data Engineering* **8**(2), 195–210 (1996).

[59] G. C. Canavos. "Probabilidad y estadística: Aplicaciones y métodos". McGraw-Hill (1998).

[60] A. Cano, J. G. Castellano, A. Masegosa y S. Moral. Aplicación de un modelo naïve bayes gaussiano con selección de variables al análisis de datos de expresión genética. En "Actas de las VI Jornadas de Transferencia Tecnológica de Inteligencia Artificial (TTIA'2005). I Congreso Español de Informática (CEDI 2005)", págs. 81–88. Thomson (2005).

[61] A. Cano, J. G. Castellano, A. R. Masegosa y S. Moral. Application of a selective Gaussian naive bayes model for diffuse large-B-cell lymphoma classification. En P. Lucas, editor, "Proceedings of the 2nd European Workshop on Probabilistic Graphical Models(PGM'04)", págs. 33–40 (2004).

[62] A. Cano, J. G. Castellano, A. R. Masegosa y S. Moral. Selective gaussian naive bayes model for diffuse large-B-cell lymphoma classification: Some improvements in preprocessing and variable elimination. En L. Godó, editor, "Symbolic and Quantitative Approaches to Reasoning with Uncertainty ECSQARU2005", vol. 3571 de "Lectures Notes in Artificial Intelligence", págs. 908–920. Springer-Verlag (2005).

[63] E. Castillo, J. M. Gutiérrez y A. S. Hadi. "Sistemas expertos y modelos de redes probabilísticas". Academia de Ingeniería (1997).

[64] J. H. Chang, K. B. Hwang, S. J. Oh y B. T. Zhang. Bayesian network learning with feature abstraction for gene-drug dependency analysis. *Journal of Bioinformatics and Computational Biology* **3**(1), 61–78 (2005).

[65] J. H. Chang, K. B. Hwang y B. T. Zhang. Analysis of gene expression profiles and drug activity patterns by clustering and Bayesian network learning. En S. M. Lin y K. F. Johnson, editores, "Methods of Microarray Data Analysis II (Proceedings of CAMDA'01),", págs. 169–184. Kluwer Academic Publishers (2002).

[66] P. Cheeseman. In defense of probability. En "Proceedings of the 9th International Joint Conference on Artificial Intelligence (IJCAI-85)", págs. 1002–1009. Morgan Kaufmann Publishers (1985).

[67] P. Cheeseman y J. Stutz. Bayesian classification (AutoClass): theory and results. *Advances in knowledge discovery and data mining* págs. 153–180 (1996).

[68] J. Cheng, D. A. Bell y W. Liu. An algorithm for Bayesian belief network construction from data. En "Proceedings of Artificial Intelligence and Statistics (AI and STAT97)", págs. 83–90 (1997).

[69] J. Cheng y R. Greiner. Comparing Bayesian network classifiers. En "UAI '99: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence", págs. 101–108. Morgan Kaufmann Publishers (1999).

[70] J. Cheng y R. Greiner. Learning Bayesian belief network classifiers: Algorithms and system. En "AI '01: Proceedings of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence", págs. 141–151. Springer-Verlag (2001).

[71] J. Cheng, R. Greiner, J. Kelly, D. Bell y W. Liu. Learning Bayesian networks from data: An information-theory based approach. *Artificial Intelligence* **137**(1-2), 43–90 (2002).

[72] D. M. Chickering. A transformational characterization of equivalent Bayesian network structures. En P. Besnard y S. Hanks, editores, "UAI '95: Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence", págs. 87–98. Morgan Kaufmann (1995).

[73] D. M. Chickering. Learning equivalence classes of Bayesian network structures. En E. Horvitz y F. V. Jensen, editores, "Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence", págs. 150–157. Morgan Kaufmann (1996).

[74] D. M. Chickering. Learning equivalence classes of Bayesian-network structures. *Journal of Machine Learning Research* **2**, 445–498 (2002).

[75] D. M. Chickering, D. Geiger y D. Heckerman. Learning Bayesian networks is NP-hard. Informe técnico MSR-TR-94-17, Microsoft research (1994).

[76] D. M. Chickering, D. Geiger y D. Heckerman. Learning Bayesian networks: Search methods and experimental results. En "Proceedings of Fifth Conference on Artificial Intelligence and Statistics", págs. 112–128 (1995).

[77] R. J. Cho, M. J. Campbell, E. A. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. G. Wolfsberg, A. E. Gabrielian, D. Landsman, D. J. Lockhart y R. W. Davis. A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell* **2**(1), 65–73 (1998).

[78] C. Chow y C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory* **14**(3), 462–467 (1968).

[79] P. A. Clarke, R. te Poele, R. Wooster y P. Workman. Gene expression microarray analysis in cancer biology, pharmacology, and drug development: Progress and potential. *Biochemical Pharmacology* **62**(10), 1311–1336 (2001).

[80] G. F. Cooper y E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning* **9**, 309–347 (1992).

[81] G. F. Cooper y C. Yoo. Causal discovery from a mixture of data. En "Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)", págs. 116–125. Morgan Kaufmann Publishers (1999).

[82] M. Craven, D. Page, J. Shavlik, J. Bockhorst y J. Glasner. A probabilistic learning approach to whole-genome operon prediction. En "Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology", vol. 8, págs. 116–27 (2000).

[83] D. Dash y M. J. Druzdzel. A hybrid anytime algorithm for the construction of causal models from sparse data. En K. B. Laskey y H. Prade, editores, "UAI '99: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence", págs. 142–149. Morgan Kaufmann (1999).

[84] L. M. de Campos. Independency relationships and learning algorithms for singly connected networks. *Journal of Experimental and Theoretical Artificial Intelligence* **10**(4), 511–549 (1998).

[85] L. M. de Campos. A scoring function for learning Bayesian networks based on mutual information and conditional independence tests. *Journal of Machine Learning Research* **7**, 2149–2187 (2006).

[86] L. M. de Campos, J. M. Fernández-Luna, J. A. Gámez y J. M. Puerta. Ant colony optimization for learning Bayesian networks. *International Journal of Approximate Reasoning* **31**, 291–311 (2002).

[87] L. M. de Campos, J. M. Fernández-Luna y J. M. Puerta. Local search methods for learning Bayesian networks using a modified neighborhood in the space of dags. En F. J. Garijo, J. C. R. Santos y M. Toro, editores, "Advances in Artificial Intelligence - IBERAMIA 2002, 8th Ibero-American Conference on AI, Proceedings", vol. 2527 de "Lecture Notes in Computer Science", págs. 182–192. Springer (2002).

[88] L. M. de Campos, J. M. Fernández-Luna y J. M. Puerta. An iterated local search algorithm for learning Bayesian networks with restarts based on conditional independence tests. *International Journal of Intelligent Systems* **18**(2), 221–235 (2003).

[89] L. M. de Campos y J. F. Huete. A new approach for learning belief networks using independence criteria. *International Journal of Approximate Reasoning* **24**(1), 11–37 (2000).

[90] L. M. de Campos y J. M. Puerta. Stochastic local and distributed search algorithms for learning belief networks. En "Proceedings of the III International Symposium on Adaptive Systems: Evolutionary Computation and Probabilistic Graphical Model", págs. 109–115 (2001).

[91] B. de Finetti. Probabilism: A critical essay on the theory of probability and on the value of science. *Erkenntnis* **31** (1989).

[92] C. Debouck y P. Goodfellow. DNA microarrays in drug discovery and development. *Nature Genetics* **21**(1), 48–50 (1999).

[93] A. P. Dempster, N. M. Laird y D. B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society B* **39**, 1–38 (1977).

[94] J. L. DeRisi, V. R. Iyer y P. O. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science* **278**(5338), 680–686 (1997).

[95] Q. Diao, W. Hu, H. Zhong, J. Li, F. Xue, T. Wang y Y. Zhan. Disease gene explorer: Display disease gene dependency by combining Bayesian networks with clustering. En "3rd International IEEE Computer Society Computational Systems Bioinformatics Conference", págs. 574–575. IEEE Computer Society (2004).

[96] F. J. Díez. Aplicaciones de los modelos gráficos probabilísticos en medicina. En J. A. Gámez y J. M. Puerta, editores, "Sistemas Expertos Probabilísticos", Ciencia y Técnica num. 20, págs. 239–263. Universidad de Castilla-La Mancha, 1 ed. (1998).

[97] C. Ding y H. Peng. Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology* **3**(2), 185–205 (2005).

[98] A. Djebbari y J. Quackenbush. Seeded Bayesian networks: constructing genetic networks from microarray data. *BMC Systems Biology* **2**, 57 (2008).

[99] J. Dougherty, R. Kohavi y M. Sahami. Supervised and unsupervised discretization of continuous features. En "International Conference on Machine Learning", págs. 194–202 (1995).

[100] R. Duda y P. Hart. "Pattern classification and scene analysis". John Wiley and Sons, New York (1973).

[101] B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics* **7**, 1–26 (1979).

[102] B. Efron y R. Tibshirani. "An Introduction to the bootstrap". Chapman and Hall (1993).

[103] M. B. Eisen, P. T. Spellman, P. O. Brown y D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America (PNAS)* **95**(25), 14863–8 (1998).

[104] G. Elidan y N. Friedman. Learning hidden variable networks: The information bottleneck approach. *Journal of Machine Learning Research* **6**, 81–127 (2005).

[105] Elvira Consortium. Elvira: An environment for probabilistic graphical models. En J. Gámez y A. Salmerón, editores, "Procs. of the 1st European Workshop on Probabilistics Graphical Models (PGM 2002)", págs. 222–230 (2002).

[106] S. B. English, S. Shih, M. F. Ramoni, L. E. Smith y A. J. Butte. Use of Bayesian networks to probabilistically model and improve the likelihood of validation of microarray findings by RT-PCR. *Journal of Biomedical Informatics* **42**(2), 287–295 (2009).

[107] F. Esposito, D. Malerba y G. Semeraro. A comparative analysis of methods for pruning decision trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(5), 476–491 (1997).

[108] J. A. Falconer, B. J. Naughton, D. D. Dunlop, E. J. Roth y D. Strasser. Predicting stroke inpatient rehabilitation outcome using a classification tree approach. *Archives of Physical Medicine and Rehabilitation* **75**(6), 619 (1994).

[109] L. Fan, K.-L. Poha y P. Zhoub. A sequential feature extraction approach for naïve bayes classification of microarray data. *Expert Systems with Applications* **36**(6), 9919–9923 (2009).

[110] T. Fawcett. ROC graphs: Notes and practical considerations for researchers. Informe técnico HPL-2003-4, HP Labs, Palo Alto, USA (2004).

[111] U. Fayyad y K. Irani. Multi-valued interval discretization of continuous-valued attributes for classification learning. En "Proceeding of the 13th International joint Conference on Artificial Inteligence", págs. 1022–1027. Morgan Kaufmann (1993).

[112] G. F.Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence* **42**, 393–405 (1990).

[113] P. E. File, P. I. Dugard y A. S. Houston. Evaluation of the use of induction in the develeopment of a medical expert system. *Computers and Biomedical Research* **27**(5), 383–395 (1994).

[114] E. Fix y J. J. L. Hodges. Discriminatory analysis: nonparametric discrimination: consistency properties. Informe técnico Project 21-49-004, Report Number 4, USAF School of Aviation Medicine (1951).

[115] N. Friedman. The Bayesian structural EM algorithm. En G. F. Cooper y S. Moral, editores, "Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)", págs. 129–138. Morgan Kaufmann Publishers (1998).

[116] N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science* **303**(5659), 799–805 (2004).

[117] N. Friedman, I. Nachman y D. Peér. Learning Bayesian network structure from massive datasets: The "Sparse Candidate" algorithm. En "UAI '99: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence", págs. 206–215 (1999).

[118] N. Friedman, D. Geiger y M. Goldszmidt. Bayesian networks classifiers. *Machine Learning* **29**, 131–163 (1997).

[119] N. Friedman y M. Goldszmidt. Learning Bayesian networks with local structure. En E. Horvitz y F. V. Jensen, editores, "Proceedings of the 12th Annual Conference on Uncertainty in Artificial Intelligence (UAI-96)", págs. 252–262. Morgan Kaufmann Publishers (1996).

[120] N. Friedman, M. Goldszmidt y A. J. Wyner. Data analysis with Bayesian networks: A bootstrap approach. En "Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–99)", págs. 206–215. Morgan Kaufmann (1999).

[121] N. Friedman y D. Koller. Being Bayesian about network structure. En "UAI '00: Proceedings of the 16th Conference on Uncertainty in

Artificial Intelligence", págs. 201–210. Morgan Kaufmann Publishers Inc. (2000).

[122] N. Friedman, M. Linial, I. Nachman y D. Pe'er. Using Bayesian networks to analyze expression data. *Journal of Computational Biology* **7**(3-4), 601–620 (2000).

[123] G. Gamberoni, E. Lamma, F. Riguzzi, S. Storari y S. Volinia. Bayesian networks learning for gene expression datasets. En A. F. Famili, J. Kok, J. M. Peña, A. Siebes y A. J. Feelders, editores, "Advances in Intelligent Data Analysis VI, 6th International Symposium on Intelligent Data Analysis, IDA 2005, Proceedings", vol. 3646 de "Lecture Notes in Computer Science", págs. 109–120. Springer (2005).

[124] J. A. Gámez. "Inferencia abductiva en redes causales". Tesis Doctoral, Departamento de Ciencias de la Computación e I.A. Escuela Técnica Superior de Ingeniería Informática. Universidad de Granada (1998).

[125] A. P. Gasch, P. T. Spellman, C. M. Kao, O. C. Harel, M. B. Eisen, G. Storz, D. Botstein y P. O. Brown. Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell* **11**(12), 4241–4257 (2000).

[126] D. Geiger y D. Heckerman. Learning Gaussian networks. En R. L. de Mántaras y D. Poole, editores, "UAI '94: Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence", págs. 235–243. Morgan Kaufmann Publishers (1994).

[127] D. Geiger y D. Heckerman. A characterization of the Dirichlet distribution with application to learning Bayesian networks. En P. Besnard y S. Hanks, editores, "UAI '95: Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence", págs. 196–207. Morgan Kaufmann (1995).

[128] D. Geiger y D. Heckerman. Knowledge representation and inference in similarity networks and Bayesian multinets. *Artificial Intelligence* **82**, 45–74 (1996).

[129] D. Geiger y D. Heckerman. A characterization of the Dirichlet distribution through global and local parameter independence. *Annals of Statistics* **25**, 1344–1369 (1997).

[130] D. Geiger, A. Paz y J. Pearl. Learning simple causal structures. *Knowledge acquisition as modeling* **8**, 231–247 (1993).

[131] O. Gevaert, F. D. Smet, D. Timmerman, Y. Moreau y B. D. Moor. Predicting the prognosis of breast cancer by integrating clinical and microarray data with Bayesian networks. *Bioinformatics* **22**(14), e184–e190 (2006).

[132] O. Gevaert, S. Van Vooren y B. de Moor. Integration of microarray and textual data improves the prognosis prediction of breast, lung and ovarian cancer patients. *Pacific Symposium on Biocomputing* págs. 279–290 (2008).

[133] Z. Ghahramani. Learning dynamic Bayesian networks. En "Adaptive Processing of Sequences and Data Structures, International Summer School on Neural Networks", vol. 1387 de "Lecture Notes in Computer Science", págs. 168–197. Springer-Verlag (1998).

[134] D. K. Gifford. Blazing pathways through genetic mountains. *Science* **293**, 2049–2051 (2001).

[135] S. B. Gillispie y C. Lemieux. Enumerating Markov equivalence classes of acyclic digraph models. En "UAI '01: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence", págs. 171–177. Morgan Kaufmann Publishers Inc. (2001).

[136] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield y E. S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **286**, 531–537 (1999).

[137] I. J. Good. "The estimation of probabilities". The MIT Press (1965).

[138] T. Graepel, M. Burger y K. Obermayer. Self-organizing maps: Generalizations and new optimization techniques. *Neurocomputing* **20**, 173–190 (1998).

[139] A. J. F. Griffiths, J. H. Miller, D. T. Suzuki, R. C. Lewontin y W. M. Gelbart. "Genética". Interamericana, España, 7 ed. (2002).

[140] M. Grzegorczyk, D. Husmeier, K. D. Edwards, P. Ghazal y A. J. Millar. Modelling non-stationary gene regulatory processes with a non-homogeneous Bayesian network and the allocation sampler. *Bioinformatics* **24**(18), 2071–2078 (2008).

[141] M. A. Hall y L. A. Smith. Feature subset selection: a correlation based filter approach. En "International Conference on Neural Information Processing and Intelligent Information Systems", págs. 855–858. Springer (1997).

[142] A. J. Hartemink, D. K. Gifford, T. S. Jaakkola y R. A. Young. Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. *Pacific Symposium on Biocomputing* págs. 422–33 (2001).

[143] A. J. Hartemink, D. K. Gifford, T. S. Jaakkola y R. A. Young. Bayesian methods for elucidating genetic regulatory networks. *IEEE Intelligent Systems* **17**(2), 37–43 (2002).

[144] A. J. Hartemink, D. K. Gifford, T. S. Jaakkola y R. A. Young. Combining location and expression data for principled discovery of genetic regulatory network models. *Pacific Symposium on Biocomputing* págs. 437–49 (2002).

[145] T. J. Hastie y R. J. Tibshirani. "Generalized additive models". London: Chapman & Hall (1990).

[146] S. Hautaniemi, H. Edgren, P. Vesanen, M. Wolf, A. K. Järvinen, O. Yli-Harja, J. Astola, O. P. Kallioniemi y O. Monni. A novel strategy for microarray quality control using Bayesian networks. *Bioinformatics* **19**(16), 2031–2038 (2003).

[147] D. Heckerman. "Probabilistic similarity networks". ACM Doctoral dissertation award series, MIT Press (1991).

[148] D. Heckerman. A tutorial on learning with Bayesian networks. Informe técnico MSR-TR-95-06, Microsoft Research, Advanced Technology Division (1995).

[149] D. Heckerman. Bayesian networks for knowledge discovery. *Advances in knowledge discovery and data mining* págs. 273–305 (1996).

[150] D. Heckerman, D. Geiger y D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning* **20**(3), 197–243 (1995).

[151] I. Hedenfalk, Y. C. D. Duggan, M. Radmacher, M. Bittner, R. Simon, P. Meltzer, B. Gusterson, M. Esteller, O. Kallioniemi, B. Wilfond, A. Borg y J. Trent. Gene-expression profiles in hereditary breast cancer. *The new england journal of medicine* **344**(9), 539–548 (2001).

[152] P. Helman, R. Veroff, S. R. Atlas y C. Willman. A Bayesian network classification methodology for gene expression data. *Journal of Computational Biology* **11**(4), 581 – 615 (2004).

[153] L. D. Hernández. Algoritmos de propagación i: Métodos exactos. En J. A. Gámez y J. M. Puerta, editores, "Sistemas Expertos Probabilísticos", Ciencia y Técnica num. 20, págs. 41–64. Universidad de Castilla-La Mancha, 1 ed. (1998).

[154] E. Herskovits y G. F. Cooper. Kutató: An entropy-driven system for the construction of probabilistic expert systems from databases. En "Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence", págs. 54–62 (1990).

[155] A. Hopkins, J. Johnson, S. Lahart, D. McLaughlin, C. W. Warner, M. Q. Wright y J. D. Maton. "Cells, building blocks of life". Pearson Prentice Hall (1997).

[156] K. Hornik, C. Buchta y A. Zeileis. Open-source machine learning: R meets Weka. *Computational Statistics* **24**(2), 225–232 (2009).

[157] Y. Huang, J. Wang, J. Zhang, M. Sanchez y Y. Wang. Bayesian inference of genetic regulatory networks from time series microarray data using dynamic Bayesian networks. *Journal of Multimedia* **2**(3), 46–56 (2007).

[158] Z. Huang, J. Li, H. Su, G. S. Watts y H. Chen. Large-scale regulatory network analysis from microarray data: Modified Bayesian network learning and association rule mining. *Decision Support Systems* **43**(4), 1207–1225 (2007).

[159] T. R. Hughes, M. J. Marton, A. R. Jones, C. J. Roberts, R. Stoughton, C. D. Armour, H. A. Bennett, E. Coffey, H. Dai, Y. D. He, M. J. Kidd, A. M. King, M. R. Meyer, D. Slade, P. Y. Lum, S. B. Stepaniants, D. D. Shoemaker, D. Gachotte, K. Chakraburtty, J. Simon, M. Bard y S. H. Friend. Functional discovery via a compendium of expression profiles. *Cell* **102**(1), 109–126 (2000).

[160] E. Hun, J. Marin y P. Stone. Experiments in induction. *Academic Press* (1966).

[161] D. Husmeier. Reverse engineering of genetic networks with Bayesian networks. *Biochemical Society Transactions* **31**(Pt 6), 1516–8 (2003).

[162] D. Husmeier. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics* **19**(17), 2271–82 (2003).

[163] K. B. Hwang, D. Y. Cho, S. W. Park, S. D. Kim y B. T. Zhang. Applying machine learning techniques to analysis of gene expression data: Cancer diagnosis. En S. M. Lin y K. F. Johnson, editores, "Methods of Microarray Data Analysis (Proceedings of CAMDA'00),", págs. 167–182. Kluwer Academic Publishers (2002).

[164] K. B. Hwang y B. T. Zhang. Bayesian model averaging of Bayesian network classifiers over multiple node-orders: application to sparse datasets. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* **35**(6), 1302–1310 (2005).

[165] J. G. Ibrahim, M. H. Chen y R. J. Gray. Bayesian models for gene expression with DNA microarray data. *Journal of the American Statistical Association* **97**, 88–99 (2002).

[166] S. Imoto, T. Goto y S. Miyano. Estimation of genetic networks and functional structures between genes by using Bayesian network and

nonparametric regression. En "Pacific Symposium on Biocomputing", vol. 7, págs. 175–186 (2002).

[167] S. Imoto, T. Higuchi, T. Goto y S. Miyano. Error tolerant model for incorporating biological knowledge with expression data in estimating. *Statistical Methodology* **3**(1), 1–16 (2006).

[168] S. Imoto, T. Higuchi, T. Goto, K. Tashiro, S. Kuhara y S. Miyano. Combining microarrays and biological knowledge for estimating gene networks via Bayesian networks. En "2nd IEEE Computer Society Bioinformatics Conference (CSB 2003)", págs. 104–113. IEEE Computer Society (2003).

[169] S. Imoto, T. Higuchi, T. Goto, K. Tashiro, S. Kuhara y S. Miyano. Combining microarrays and biological knowledge for estimating gene networks via Bayesian networks. *Journal of Bioinformatics and Computational Biology* **2**(1), 77–98 (2004).

[170] S. Imoto, S. Kim, T. Goto, S. Miyano, S. Aburatani, K. Tashiro y S. Kuhara. Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network. *Journal of Bioinformatics and Computational Biology* **1**(2), 231–52 (2003).

[171] S. Imoto, H. Shimodaira, S. Kim, S. Aburatani, K. Tashiro, S. Kuhara y S. Miyano. Bootstrap analysis of gene networks based on Bayesian networks and nonparametric regression. En "Proceedings of the International Conference on Genome Informatics", vol. 13, págs. 369–370 (2002).

[172] S. Imoto, K. Sunyong, T. Goto, S. Aburatani, K. Tashiro, S. Kuhara y S. Miyano. Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network. En "CSB '02: Proceedings of the IEEE Computer Society Conference on Bioinformatics", pág. 219. IEEE Computer Society (2002).

[173] I. Inza, P. Larrañaga, R. Blanco y A. J. Cerrolaza. Filter versus wrapper gene selection approaches in DNA microarray domains. *Artificial Intelligence in Medicine* **31**(2), 91–103 (2004).

[174] I. Inza, P. Larrañaga, R. Etxeberria y B. Sierra. Feature subset selection by Bayesian network-based optimization. *Artificial Intelligence* **123**(1-2), 157–184 (2000).

[175] I. Inza, B. Sierra, R. Blanco y P. Larrañaga. Gene selection by sequential search wrapper approaches in microarray cancer class prediction. *Journal of Intelligent and Fuzzy Systems* **12**(1), 25–34 (2002).

[176] G. H. John, R. Kohavi y K. Pfleger. Irrelevant features and the subset selection problem. En "International Conference on Machine Learning", págs. 121–129 (1994).

[177] G. Judmaier, P. Meyersbach, G. Weiss, H. Wachter y G. Reibnegger. The role of neopterin in assessing disease activity in Crohn's disease: Classification and regression trees. *The American Journal of Gastroenterology* **88**, 706–711 (1993).

[178] T. Kamimura, H. Shimodaira, S. Imoto, S. Kim, K. Tashiro, S. Kuhara y S. Miyano. Multiscale bootstrap analysis of gene networks based on Bayesian networks and nonparametric regression. En "Proceedings of the International Conference on Genome Informatics", vol. 14, págs. 350–351 (2002).

[179] M. Kanehisa, M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, T. Tokimatsu y Y. Yamanishi. KEGG for linking genomes to life and the environment. *Nucleic Acids Research* **36**, 480–484 (2008).

[180] S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology* **22**(3), 437–467 (1969).

[181] E. J. Keogh y M. J. Pazzani. Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches. En "Proceedings of 7th International Workshop on Artificial Intelligence and Statistics", págs. 225–230 (1999).

[182] E. J. Keogh y M. J. Pazzani. Learning the structure of augmented Bayesian classifiers. *International Journal on Artificial Intelligence Tools* **11**(4), 587–601 (2002).

[183] J. Khan, J. S. Wei, M. Ringnér, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson y P. S. Meltzer. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine* **7**, 673–679 (2001).

[184] A. B. Khodursky, B. J. Peter, N. R. Cozzarelli, D. Botstein, P. O. Brown y C. Yanofsky. DNA microarray analysis of gene expression in response to physiological and genetic changes that affect tryptophan metabolism in Escherichia coli. *Proceeding National Academy of Sciences of USA* **97**(22), 12170–5 (2000).

[185] H. Kim, G. H. Golub y H. Park. Missing value estimation for DNA microarray gene expression data: local least squares imputation. *Bioinformatics* **21**(2), 187–198 (2005).

[186] S. Kim, S. Imoto y S. Miyano. Dynamic Bayesian network and nonparametric regression model for inferring gene networks. En "Proceedings of the International Conference on Genome Informatics", vol. 13, págs. 371–372 (2002).

[187] S. Kim, S. Imoto y S. Miyano. Dynamic Bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data. En C. Priami, editor, "Computational Methods in Systems Biology, First International Workshop, CMSB 2003", vol. 2602 de "Lecture Notes in Computer Science", págs. 104–113. Springer (2003).

[188] S. Kim, S. Imoto y S. Miyano. Inferring gene networks from time series microarray data using dynamic Bayesian networks. *Briefings in Bioinformatics* **4**(3), 228–35 (2003).

[189] S. Kim, S. Imoto y S. Miyano. Dynamic Bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data. *Biosystems* **75**(1-3), 57–65 (2004).

[190] L. Klebanov y A. Yakovlev. How high is the level of technical noise in microarray data? *Biology Direct* **2**, 2–9 (2007).

[191] S. Knudsen. "Cancer diagnostics with DNA microarrays". John Wiley & Sons, Inc. (2006).

[192] Y. Ko, C. Zhai y S. L. Rodriguez-Zas. Inference of gene pathways using Gaussian mixture models. En "BIBM '07: Proceedings of the 2007 IEEE International Conference on Bioinformatics and Biomedicine", págs. 362–367. IEEE Computer Society (2007).

[193] T. Kocka y R. Castelo. Improved learning of Bayesian networks. En "Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence (UAI-01)", págs. 269–276. Morgan Kaufmann Publishers (2001).

[194] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. En "Fourteenth International Joint Conference on Artificial Intelligence", págs. 1137–1145 (1995).

[195] R. Kohavi. Scaling up the accuracy o naïve-bayes classifier: A decision tree hybrid. *Proceeedings of the Second International Conference on Knowledge Discovery and Data Mining.* (1996).

[196] R. Kohavi. "Wrappers for performance enhancement and oblivious decision graphs". Tesis Doctoral, Stanford University, Stanford, CA, USA (1996).

[197] R. Kohavi y G. H. John. Wrappers for feature subset selection. *Artificial Intelligence* **97**(1-2), 273–324 (1997).

[198] R. Kohavi, D. Sommerfield y J. Dougherty. Data mining using MLC++: A machine learning library in C++. *International Journal on Artificial Intelligence Tools* **6**(4), 537–566 (1997).

[199] P. Kokol, M. Mernik, J. Zavrsnik y K. Kancler. Decision trees based on automatic learning and their use in cardiology. *Journal of Medical Systems* **18**(4), 201 (1994).

[200] A. N. Kolmogorov. "Foundations of the theory of probability". Chelsea Publishing Company, New York, 2 ed. (1956).

[201] I. Kononenko. Semi-naive Bayesian classifier. En "European Working Session on Learning on Machine Learning", págs. 206–219 (1991).

[202] I. Kononenko. Inductive and Bayesian learning in medical diagnosis. *Applied Artificial Intelligence* **4**(7), 317–337 (1993).

[203] P. Lachenbruch y A. Mickey. Estimation of error rates in discriminant analysis. *Technometrics* **10**, 1–11 (1968).

[204] W. Lam y F. Bacchus. Learning Bayesian belief networks: An approach based on the MDL principle. *Computational Intelligence* **10**, 269–294 (1994).

[205] P. Langely y S. Sage. Induction of selective Bayesian classifiers. *Proceeding of the Tenth Conference on Uncertainty in Artificial Intelligence* págs. 399–406 (1998).

[206] P. Langley, W. Iba y K. Thompson. An analysis of Bayesian classifiers. En "National Conference on Artificial Intelligence", págs. 223–228 (1992).

[207] P. Larrañaga, C. M. H. Kuijpers, R. H. Murga y Y. Yurramendi. Learning Bayesian network structures by searching for the best ordering with genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics* **26**, 487–493 (1996).

[208] P. Larrañaga y J. A. Lozano. "Estimation of distribution algorithms: A new tool for evolutionary computation". Genetic Algorithms and Evolutionary Computation. Kluwer Academic Publishers (2001).

[209] P. Larrañaga, M. Poza, Y. Yurramendi, R. H. Murga y C. M. H. Kuijpers. Structure learning of Bayesian networks by genetic algorithms: A performance analysis of control parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(9), 912–926 (1996).

[210] P. Larsen, E. Almasri, G. Chen y Y. Dai. A statistical method to incorporate biological knowledge for generating testable novel gene regulatory interactions from microarray experiments. *BMC Bioinformatics* **8**, 317 (2007).

[211] S. L. Lauritzen. Propagation of probabilities, means and variances in mixed graphical association models. *Journal of the American Statistical Association* **87**, 1098–1108 (1992).

[212] S. L. Lauritzen y D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert

systems (with discussion). *Journal of the Royal Statistical Society, Series B* **50**, 157–224 (1988).

[213] S. L. Lauritzen y N. Wermuth. Graphical models for associations between variables, some of which are qualitative and some quantitative. *Annals of Statistics* **17**(1), 31–57 (1998).

[214] P. Le Phillip, A. Bahl y L. H. Ungar. Using prior knowledge to improve genetic network reconstruction from microarray data. *In Silico Biology* **4**(3), 335–353 (2004).

[215] R. D. Leclerc. Survival of the sparsest: robust gene networks are parsimonious. *Molecular Systems Biology* **4**(213) (2008).

[216] T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J.-B. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford y R. A. Young. Transcriptional regulatory networks in Saccharomyces cerevisiae. *Science* **298**(5594), 799–804 (2002).

[217] J. Li, H. Liu y L. Wong. Mean-entropy discretized features are effective for classifying high-dimensional biomedical data. En M. J. Zaki, J. T.-L. Wang y H. Toivonen, editores, "Proceedings of the 3nd ACM SIGKDD Workshop on Data Mining in Bioinformatics (BIOKDD 2003)", págs. 17–24 (2003).

[218] D. V. Lindley. "Introduction to probability and statistics from a Bayesian viewpoint". Cambridge University Press (1965).

[219] H. Liu y M. Hiroshi. "Feature selection for knowledge discovery and data mining". Kluwer Academic Publishers (1998).

[220] A. D. Long, H. J. Mangalam, B. Y. P. Chan, L. Tolleri, G. W. Hatfield y P. Baldi. Improved statistical inference from DNA microarray data using analysis of variance and a Bayesian statistical framework. Analysis of global gene expression in Escherichia coli K12. *Journal of Biological Chemistry* **276**(23), 19937–19944 (2001).

[221] M. López, P. Mallorquín y M. Vega. "Microarrays y biochips de ADN. Informe de Vigilancia Tecnológica." Genoma España / CIBT-FGUAM (2002).

[222] P. J. F. Lucas. Restricted Bayesian network structure learning. En J. Gámez y A. Salmerón, editores, "Procs. of the 1st European Workshop on Probabilistics Graphical Models (PGM 2002)", págs. 117–126 (2002).

[223] P. J. F. Lucas, L. C. van der Gaag y A. Abu-Hanna. Bayesian networks in biomedicine and health-care. *Artificial Intelligence in Medicine* **30**(3), 201–214 (2004).

[224] D. Madigan, S. Andersson, M. Perlman y C. Volinsky. Bayesian model averaging and model selection for Markov equivalence classes of acyclic digraphs. En "Communications in Statistics: Theory and Methods", págs. 2493–2519 (1996).

[225] D. Madigan y A. E. Raftery. Model selection and accounting for model uncertainty in graphical models using occam's window. *Journal of the American Statistical Association* **89**(428), 1535–1546 (1994).

[226] F. Markowetz y R. Spang. Inferring cellular networks - a review. *BMC Bioinformatics* **8**(6) (2007).

[227] A. R. Masegosa. "Model of Supervised Classification: Applications to Genomics and Information Retrieval." Tesis Doctoral, Granada University (2009).

[228] H. H. McAdams y A. Arkin. Stochastic mechanisms in gene expression. *PNAS Proceedings of the National Academy of Sciences of the United States of America* **94**(3), 814–819 (1997).

[229] D. McKenzie, P. McGorry, C. Wallace, L. H. Low, D. Copolov y B. Singh. Constructing a minimal diagnostic decision tree. *Methods of Information in Medicine* **32**(2), 161–166 (1993).

[230] C. Meek. Causal inference and causal explanation with background knowledge. En "Proceedings of the 11th Annual Conference on Uncertainty in Artificial Intelligence (UAI-95)", págs. 403–410. Morgan Kaufmann Publishers (1995).

[231] M. Meila y D. Heckerman. An experimental comparison of several clustering and initialization methods. En G. F. Cooper y S. Moral, editores, "Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)", págs. 386–395. Morgan Kaufmann (1998).

[232] H. W. Mewes, D. Frishman, U. Güldener, G. Mannhaupt, K. F. X. Mayer, M. Mokrejs, B. Morgenstern, M. Münsterkötter, S. Rudd y B. Weil. MIPS: a database for genomes and protein sequences. *Nucleic Acids Research* **30**(1), 31–34 (2002).

[233] T. Michoel, S. Maere, E. Bonnet, A. Joshi, Y. Saeys, T. Van den Bulcke, K. Van Leemput, P. van Remortel, M. Kuiper, K. Marchal y Y. Van de Peer. Validating module network learning algorithms using simulated data. *BMC Bioinformatics* **8**(2) (2007).

[234] J. Mingers. Expert systems - rule induction with statistical data. *Journal of the Operational Research Society* págs. 39–47 (1987).

[235] S. Miyano, R. Yamaguchi, Y. Tamada, Masao Nagasaki y S. Imoto. Gene networks viewed through two models. En S. Rajasekaran, editor, "Bioinformatics and Computational Biology, First International Conference, BICoB 2009. Proceedings", vol. 5462 de "Lecture Notes in Computer Science", págs. 54–66. Springer (2009).

[236] E. Moler, M. Chow y I. Mian. Analysis of molecular profile data using generative and discriminative methods. *Physiological Genomics* **4**(2), 109–126 (2000).

[237] M. Mramor, G. Leban, J. Demsar y B. Zupan. Visualization-based cancer microarray data classification analysis. *Bioinformatics* (2007).

[238] P. Munteanu y D. Cau. Efficient score-based learning of equivalence classes of Bayesian networks. En D. A. Zighed, H. J. Komorowski y J. M. Zytkow, editores, "Principles of Data Mining and Knowledge Discovery, 4th European Conference, PKDD 2000, Lyon, France, September 13-16, 2000, Proceedings", vol. 1910 de "Lecture Notes in Computer Science", págs. 96–105. Springer (2000).

[239] K. P. Murphy. "Dynamic Bayesian networks: Representation, inference and learning". Tesis Doctoral, UC Berkeley, Computer Science Division (2002).

[240] K. P. Murphy y S. Mian. Modelling gene expression data using dynamic Bayesian networks. Informe técnico, Computer Science Division, University of California, Berkeley, CA (1999).

[241] J. W. Myers, K. B. Laskey y T. S. Levitt. Learning Bayesian networks from incomplete data with stochastic search algorithms. En K. B. Laskey y H. Prade, editores, "UAI '99: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence", págs. 476–485. Morgan Kaufmann (1999).

[242] N. Nariai, S. Kim, S. Imoto y S. Miyano. Using protein-protein interactions for refining gene networks estimated from microarray data by Bayesian networks. *Pacific Symposium on Biocomputing* págs. 336–47 (2004).

[243] N. Nariai, Y. Tamada, S. Imoto y S. Miyano. Estimating gene regulatory networks and protein–protein interactions of Saccharomyces cerevisiae from multiple genome-wide data. *Bioinformatics* **21**(2), 206–212 (2005).

[244] T. Niblett y I. Bratko. Learning decision rules in noisy domains. En "Proceedings of Expert Systems '86, The 6Th Annual Technical Conference on Research and development in expert systems III", págs. 25–34. Cambridge University Press (1986).

[245] W. Nicholas, M. F. Usama y S. Djorgovski. Initial galaxy counts from digitized poss-II. *Astronomical Journal* **109**(6), 2401 (1995).

[246] J. D. Nielsen, T. Kocka y J. M. Peña. On local optima in learning Bayesian networks. En C. Meek y U. Kjærulff, editores, "UAI '03, Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence", págs. 435–442. Morgan Kaufmann (2003).

[247] J. Niemi. Accuracy of the Bayesian network algorithms for inferring gene regulatory networks. Independent research projects in applied mathematics Mat-2.108, Helsinky University of Technology (2007).

[248] N. Novershtern, Z. Itzhaki, O. Manor, N. Friedman y N. Kaminski. A functional and regulatory map of asthma. *American Journal of Respiratory Cell and Molecular Biology.* **38**(3), 324–336 (2008).

[249] K. Numata, S. Imoto y S. Miyano. A structure learning algorithm for inference of gene networks from microarray gene expression data using Bayesian networks. En "Proceedings of the 7th IEEE International Conference on Bioinformatics and Bioengineering", págs. 1280–1284. IEEE (2007).

[250] K. Olesen. Causal probabilistic networks with both discrete and continuous variables. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **15**(3), 275–279 (1993).

[251] I. M. Ong, J. D. Glasner y D. Page. Modelling regulatory pathways in E. coli from time series expression profiles. *Bioinformatics* **18**(1), S241–8 (2002).

[252] I. M. Ong y D. Page. Inferring regulatory pathways in E. coli using dynamic Bayesian networks. Informe técnico 1426, University of Wisconsin-Madison (2001).

[253] I. M. Ong y D. Page. Learnability of dynamic Bayesian networks from time series microarray data. Informe técnico 1514, University of Wisconsin-Madison (2004).

[254] A. Osareh y B. Shadgar. Classification and diagnostic prediction of cancers using gene microarray data analysis. *Journal of Applied Sciences* **9**(3), 459–468 (2009).

[255] S. Ott, A. Hansen, S. Y. Kim y S. Miyano. Superiority of network motifs over optimal networks and an application to the revelation of gene network evolution. *Bioinformatics* **21**(2), 227–238 (2005).

[256] S. Ott, S. Imoto y S. Miyano. Finding optimal models for small gene networks. En R. B. Altman, A. K. Dunker, L. Hunter, T. A. Jung y T. E. Klein, editores, "Pacific Symposium on Biocomputing", págs. 557–567. World Scientific (2004).

[257] S. Ott y S. Miyano. Finding optimal gene networks using biological constraints. En "Proceedings of the International Conference on Genome Informatics", vol. 14, págs. 124–133 (2003).

[258] R. Palacios, J. Goni, I. Martinez-Forero, J. Iranzo, J. Sepulcre, I. Melero y P. Villoslada. A network analysis of the human T-cell activation gene network identifies jagged1 as a therapeutic target for autoimmune diseases. *Public Library of Sciencie (PLoS) ONE* **2**(11) (2007).

[259] H. S. Park, S. H. Yoo y S. B. Cho. Forward selection method with regression analysis for optimal gene selection in cancer classification. *International Journal of Computer Mathematics* **84**(5), 653–667 (2007).

[260] M. J. Pazzani. Searching for dependencies in Bayesian classifiers. En "Artificial Intelligence and Statistics IV", Lecture Notes in Statistics. Springer-Verlag (1997).

[261] J. Pearl. Fusion, propagation and structuring in belief networks. *Artificial Intelligence* **29**, 241–288 (1986).

[262] J. Pearl. Distributed revision of composite beliefs. *Artificial Intelligence* **33**, 173–215 (1987).

[263] J. Pearl. "Probabilistic reasoning in intelligent systems: Networks of plausible inference". Morgan Kaufmann Publishers Inc. (1988).

[264] E. Peeling y A. Tucker. Consensus gene regulatory networks: combining multiple microarray gene expression datasets. En "COMPLIFE 2007: The Third International Symposium on Computational Life Science. AIP Conference Proceedings,", vol. 940, págs. 38–49 (2007).

[265] D. Pe'er, A. Regev, G. Elidan y N. Friedman. Inferring subnetworks from perturbed expression profiles. *Bioinformatics* **17**(1), 215–224 (2001).

[266] M. Pelikan, D. E. Goldberg y K. Sastry. Bayesian optimization algorithm, decision graphs, and Occam's razor. Illigal report no. 2000020, Illinois Genetic Algorithms Laboratory (2000).

[267] J. M. Peña. Learning and validating Bayesian network models of genetic regulatory networks. En P. Lucas, editor, "Proceedings of the 2nd European Workshop on Probabilistic Graphical Models(PGM'04)", págs. 161–168 (2004).

[268] J. M. Peña, J. Björkegren y J. Tegnér. Growing Bayesian network models of gene networks from seed genes. *Bioinformatics* **21**(2), 224–229 (2005).

[269] J. M. Peña, J. Björkegren y J. Tegnér. Learning and validating Bayesian network models of gene networks. En "Advances in Probabilistic Graphical Models", págs. 359–375. Springer-Verlag (2007).

[270] J. M. Peña, J. A. Lozano y P. Larrañaga. Learning recursive Bayesian multinets for data clustering by means of constructive induction. *Machine Learning, 47* págs. 63–89 (2002).

[271] J. M. Peña, J. A. Lozano y P. Larrañaga. Unsupervised learning of Bayesian networks via estimation of distribution algorithms: An application to gene expression data clustering. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **12**, 63–82 (2004).

[272] M. A. Peot. Geometric implications of the naïve bayes assumption. *Uncertainty in Artificial Intelligence. Proceeedings of the Twelfth conference.* págs. 414–419 (1996).

[273] F. Pernkopf y P. O'Leary. Floating search algorithm for structure learning of bayesian network classifiers. *Pattern Recognition Letters* **24**(15), 2839–2848 (2003).

[274] B. E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet y F. d'Alché Buc. Gene networks inference using dynamic Bayesian networks. *Bioinformatics* **19**(2), 138–148 (2003).

[275] S. L. Pomeroy, P. Tamayo, M. Gaasenbeek, L. M. Sturla, M. Angelo, M. E. McLaughlin, J. Y. H. Kim, L. C. Goumnerova, P. M. Black, C. Lau, J. C. Allen, D. Zagzag, J. M. Olson, T. Curran, C. Wetmore y J. Biegel. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature* **415**, 436 – 42 (2002).

[276] F. Provost y P. Domingos. Tree induction for probability-based ranking. *Machine Learning* **52**(3) (2003).

[277] J. R. Quinlan. Discovering rules by induction from large collection of examples. *Knowledge-base systems in the Micro Electronic Age* págs. 168–201 (1979).

[278] J. R. Quinlan. Simplifying decision trees. *International Journal of Man-Machine Studies* **27**(3), 221–234 (1987).

[279] J. R. Quinlan. "C4.5: Programs for machine learning." Morgan Kaufmann Publishers Inc. (1993).

[280] R Development Core Team. "R: A Language and Environment for Statistical Computing". R Foundation for Statistical Computing, Vienna, Austria (2009). ISBN 3-900051-07-0.

[281] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C. H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. P. Mesirov, T. Poggio, W. Gerald, M. Loda, E. S. Lander y T. R. Golub. Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences of the United States of America* **98**, 15149 – 54 (2001).

[282] M. Ramoni y P. Sebastiani. Learning Bayesian networks from incomplete databases. En "Proceedings of the 13th Annual Conference on Uncertainty in Artificial Intelligence (UAI-97)", págs. 401–408. Morgan Kaufmann Publishers (1997).

[283] Y. Rao, Y. Lee, D. Jarjoura, A. Ruppert, C. gong Liu, J. Hsu y J. Hagan. A comparison of normalization techniques for microRNA microarray data. *Statistical Applications in Genetics and Molecular Biology* **7**(1), 22 (2008).

[284] E. Ravasz, A. L. Somera, D. A. Mongru, Z. Oltvai y A. L. Barabási. Hierarchical organization of modularity in metabolic networks. *Science* **297**(5586), 1551–5 (2002).

[285] S. Raychaudhuri, J. M. Stuart y R. B. Altman. Principal components analysis to summarize microarray experiments: application

to sporulation time series. *Pacific Symposium on Biocomputing* págs. 455–66 (2000).

[286] G. Rebane y J. Pearl. The recovery of causal poly-trees from statistical data. En "UAI '87: Proceedings of the Third Annual Conference on Uncertainty in Artificial Intelligence", págs. 175–182. Elsevier (1987).

[287] B. Ren, F. Robert, J. J. Wyrick, O. Aparicio, E. G. Jennings, I. Simon, J. Zeitlinger, J. Schreiber, N. Hannett, E. Kanin, T. L. Volkert, C. J. Wilson, S. P. Bell y R. A. Young. Genome-wide location and function of DNA binding proteins. *Science* **290**(5500), 2306–9 (2000).

[288] J. Rissanen. Modelling by the shortest data description. *Automatica* **14**, 465–471 (1978).

[289] J. Rissanen. Stochastic complexity and modeling. *Annals of Statistics* **14**, 1080–1100 (1986).

[290] R. W. Robinson. Counting unlabeled acyclic digraphs. En "Combinatorial mathematics V: Proceedings of the Fifth Australian Conference", vol. 622 de "Lecture notes in mathematics", págs. 28–43 (1977).

[291] M. Ronen, R. Rosenberg, B. I. Shraiman y U. Alon. Assigning numbers to the arrows: Parameterizing a gene regulation network by using accurate expression kinetics. *Proceedings of the National Academy of Sciences of the United States of America* **99**(16), 10555–60 (2002).

[292] J. Rosamond y A. Allsop. Harnessing the power of the genome in the search for new antibiotics. *Science* **287**, 1973–1976 (2000).

[293] D. T. Ross, U. Scherf, M. B. Eisen, C. M. Perou, C. Rees, P. Spellman, V. Iyer, S. S. Jeffrey, M. V. de Rijn, M. Waltham, A. Pergamenschikov, J. C. Lee, D. Lashkari, D. Shalon, T. G. Myers, J. Weinstein, D. Botstein y P. O. Brown. Systematic variation in gene expression patterns in human cancer cell lines. *Nature Genetics* **24**(3), 227–35 (2000).

[294] R. Rumí, A. Salmerón y S. Moral. Estimating mixtures of truncated exponentials in hybrid Bayesian networks. *TEST: An Official Journal of the Spanish Society of Statistics and Operations Research* **15**(2), 397–421 (2006).

[295] K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger y G. P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science* **308**(5721), 523–529 (2005).

[296] M. Sahami. Learning limited dependence Bayesian classifiers. *Second International Conference on Knowledge Discovery in Databases* págs. 335–338 (1996).

[297] A. Salmerón. Algoritmos de propagación II: Métodos de Monte Carlo. En J. A. Gámez y J. M. Puerta, editores, "Sistemas Expertos Probabilísticos", Ciencia y Técnica num. 20, págs. 65–88. Universidad de Castilla-La Mancha, 1 ed. (1998).

[298] S. Salzberg, R. Chandar, H. Forf, S. Murth y R. White. Decision trees for automated identification of cosmic-ray hits in hubble space telescope images. *Publications of the Astronomical Society of the Pacific* **107**, 1–10 (1995).

[299] C. J. Savoie, S. Aburatani, S. Watanabe, Y. Eguchi, S. Muta, S. Imoto, S. Miyano, S. Kuhara y K. Tashiro. Use of gene networks from full genome microarray libraries to identify functionally relevant drug-affected genes and gene regulation cascades. *DNA Research* **10**(1), 19–25 (2003).

[300] M. Schena, D. Shalon, R. W. Davis y P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* **270**(5235), 467–70 (1995).

[301] U. Scherf, D. T. Ross, M. Waltham, L. H. Smith, J. K. Lee, L. Tanabe, K. W. Kohn, W. C. Reinhold, T. G. Myers, D. T.Andrews, D. A. S. an d Michael B. Eisen, E. A. Sausville, Y. Pommier, D. Botstein, P. O. Brown, y J. Weinstein. A gene expression database for the molecular pharmacology of cancer. *Nature Genetics* **24**(3), 236–244 (2000).

[302] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics* **6**, 461–464 (1978).

[303] E. Segal, D. Pe'er, A. Regev, D. Koller y N. Friedman. Learning module networks. En "Proceedings Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI)", págs. 525–534 (2003).

[304] E. Segal, D. Pe'er, A. Regev, D. Koller y N. Friedman. Learning module networks. *Journal of Machine Learning Research* **6**, 557–588 (2005).

[305] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller y N. Friedman. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics* **34**(2), 166–76 (2003).

[306] E. Segal, H. Wang y D. Koller. Discovering molecular pathways from protein interaction and gene expression data. *Bioinformatics* **19**(1) (2003).

[307] I. Sen, M. P. Verdicchio, S. Jung, R. Trevino, M. Bittner y S. Kim. Context-specific gene regulations in cancer gene expression data. En "Pacific Symposium on Biocomputing", vol. 14, págs. 75–86 (2009).

[308] S. Shah y A. Kusiak. Cancer gene search with data-mining and genetic algorithms. *Computers in Biology and Medicine* **37**(2), 251–261 (2007).

[309] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal* **27**, 379–423 (1948).

[310] P. P. Shenoy y G. Shafer. Axioms for probability and belief-function propagation. En "Uncertainty in Artificial Intelligence", págs. 169–198. Morgan Kaufmann (1990).

[311] P. P. Shenoy y G. Shafer. Probability propagation. En "Annals of Mathematics and Artificial Intelligence", vol. 2, págs. 327–351 (1990).

[312] S. Shimony y E. Charniak. A new algorithm for finding MAP assignments to belief network. En "Proceedings of the 6th Annual

Conference on Uncertainty in Artificial Intelligence (UAI-91)", págs. 185–196. Elsevier Science (1991).

[313] S. Shimozono, A. Shinohara, T. Shinohara, S. Miyano, S. Kuhara y S. Arikawa. Knowledge acquisition from amino acid sequences by machine learning system BONSAI. *Transactions on Information Processing Society of Japan* **35**(10), 2009–2018 (1994).

[314] M. A. Shipp, K. Ross, P. Tamayo, A. P. Weng, J. L. Kutok, R. C. T. Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G. S. Pinkus, T. S. Ray, M. A. Koval, K. W. Last, A. Norton, T. A. Lister y J. Mesirov. Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine* **8**, 68 – 74 (2002).

[315] B. Sierra y P. Larrañaga. Predicting survival in malignant skin melanoma using Bayesian networks automatically induced by genetic algorithms. An empirical comparison between different approaches. *Artificial Intelligence in Medicine* **14**(1-2), 215–230 (1998).

[316] D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, E. S. Lander, M. Loda, P. W. Kantoff, T. R. Golub y W. R. Sellers. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell* **1**, 203 – 9 (2002).

[317] M. Singh y M. Valtorta. Construction of Bayesian network structures from data: A brief survey and an efficient algorithm. *International Journal of Approximate Reasoning* **12**(2), 111–131 (1995).

[318] D. K. Slonim, P. Tamayo, J. P. Mesirov, T. R. Golub y E. S. Lander. Class prediction and discovery using gene expression data. En "Proceedings of the fourth annual international conference on Computational molecular biology (RECOMB 2000)", págs. 263–272 (2000).

[319] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein y B. Futcher. Comprehensive identification of cell cycle-regulated genes of the

yeast Saccharomyces cerevisiae by microarray hybridization. *Molecular Biololy of the Cell* **9**, 3273–3297 (1998).

[320] P. Spirtes y C. Glymour. An algorithm for the fast recovery of sparse casual graphs. *Social Science Computer Review* **9**(1), 62–72 (1991).

[321] P. Spirtes, C. Glymour y R. Scheines. "Causation, prediction, and search". Springer-Verlag (1993).

[322] P. Spirtes, C. Glymour, R. Scheines, S. Kauffman, V. Aimale y F. Wimberly. Constructing Bayesian network models of gene expression networks from microarray data. En "Proceedings of the Atlantic Symposium on Computational Biology, Genome Information Systems and Technology" (2000).

[323] P. Spirtes y C. Meek. Learning Bayesian networks with discrete variables from data. En "Proceedings of First International Conference on Knowledge Discovery and Data Mining (KDD)", págs. 294–299. Morgan Kaufmann (1995).

[324] W. Stacklies, H. Redestig, M. Scholz, D. Walther y J. Selbig. pcaMethods—a bioconductor package providing PCA methods for incomplete data. *Bioinformatics* **23**(9), 1164–1167 (2007).

[325] H. Steck. On the use of skeletons when learning in Bayesian networks. En "Proceedings of the 16th Annual Conference on Uncertainty in Artificial Intelligence (UAI-00)", págs. 558–565. Morgan Kaufmann Publishers (2000).

[326] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society B* **36**(1), 111–147 (1974).

[327] M. P. Styczynski y G. Stephanopoulos. Overview of computational methods for the inference of gene regulatory networks. *Computers & Chemical Engineering* **29**(3), 519–534 (2005).

[328] J. Suzuki. A construction of Bayesian networks from databases based on the MDL principle. En "Proceedings of the 9th Annual Conference on Uncertainty in Artificial Intelligence", págs. 266–273. Morgan Kaufmann Publishers (1993).

[329] J. Suzuki. Learning Bayesian belief networks based on the minimum description length principle: An efficient algorithm using the B & B technique. En "International Conference on Machine Learning", págs. 462–470 (1996).

[330] J. A. Swets y R. M. Pickett. Evaluation of diagnostic systems: Methods from signal detection theory. *Medical Physics* **10**(2), 266–267 (1983).

[331] Y. Tamada, H. Bannai, S. Imoto, T. Katayama, M. Kanehisa y S. Miyano. Estimating gene networks from gene expression data and evolutionary information using Bayesian network models. En "Proceedings 15th International Conference on Genome Informatics, Posters and Software Demonstrations (GIW'04)" (2004).

[332] Y. Tamada, S. Imoto, K. Tashiro, S. Kuhara y S. Miyano. Identifying drug active pathways from gene networks estimated by gene expression data. *Genome Informatics* **16**(1), 182–191 (2005).

[333] Y. Tamada, S. Kim, H. Bannai, S. Imoto, K. Tashiro, S. Kuhara y S. Miyano. Combining gene expression data with DNA sequence information for estimating gene networks using Bayesian network model. En "Proceedings of the International Conference on Genome Informatics", vol. 14, págs. 352–353 (2003).

[334] Y. Tamada, S. Kim, H. Bannai, S. Imoto, K. Tashiro, S. Kuhara y S. Miyano. Estimating gene networks from gene expression data by combining Bayesian network model with promoter element detection. *Bioinformatics* **19**(2), 227–236 (2003).

[335] B. Thiesson, C. Meek, D. M. Chickerin y D. Heckerman. Learning mixtures of DAG models. En G. F. Cooper y S. Moral, editores, "Proceeding of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)", págs. 504–513. Morgan Kaufmann, San Francisco, CA (1998).

[336] J. Tian. A branch-and-bound algorithm for MDL learning Bayesian networks. En C. Boutilier y M. Goldszmidt, editores, "Proceedings of the 16th Annual Conference on Uncertainty in Artificial Intelligence (UAI-00)", págs. 580–588. Morgan Kaufmann Publishers (2000).

[337] I. M. Tienda-Luna, Y. Huang, Y. Yin, Diego y Carmen. Uncovering gene regulatory networks from time-series microarray data with variational Bayesian structural expectation maximization. *EURASIP Journal on Bioinformatics and Systems Biology* **2007** (2007).

[338] I. M. Tienda-Luna, Y. Yin, M. C. Carrion, Y. Huang, H. Cai, M. Sanchez y Y. Wang. Inferring the skeleton cell cycle regulatory network of malaria parasite using comparative genomic and variational Bayesian approaches. *Genetica* **132**(2), 131–142 (2007).

[339] N. Tishby, F. C. Pereira y W. Bialek. The information bottleneck method. En "The 37th annual Allerton Conference on Communication, Control, and Computing", págs. 368–377 (1999).

[340] J. B. Tobler, M.N.Molla, E. F. Nuwaysir, R. D. Green y J. W. Shavlik. Evaluating machine learning approaches for aiding probe selection for gene-expression arrays. En "Proceedings of the Tenth International Conference on Intelligent Systems for Molecular Biology ISMB (Supplement of Bioinformatics)", págs. 164–171 (2002).

[341] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein y R. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics* **17**(6), 520–525 (2001).

[342] A. Tucker, V. Vinciotti, P. A. C. 't Hoen y X. Liu. Bayesian network classifiers for time-series microarray data. En A. F. Famili, J. Kok, J. M. Peña, A. Siebes y A. J. Feelders, editores, "Advances in Intelligent Data Analysis VI, 6th International Symposium on Intelligent Data Analysis, IDA 2005, Proceedings", vol. 3646 de "Lecture Notes in Computer Science", págs. 475–485. Springer (2005).

[343] N. Turton, D. Judah, J. Riley, R. Davies, D. Lipson, J. Styles, A. Smith y T. Gant. Gene expression and amplification in breast carcinoma cells with intrinsic and acquired doxorubicin resistance. *Oncogene* **20**(11), 1300–1306 (2001).

[344] L. J. van 't Veer, H. Dai, M. J. van de Vijver, Y. D. He, A. A. Hart, M. Mao, H. L. Peterse, K. van der Kooy, M. J.

Marton, A. T. Witteveen, G. J. Schreiber, R. M. Kerkhoven, C. Roberts, P. S. Linsley, R. Bernards y S. H. Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature* **415**(6871), 530–536 (2002).

[345] T. Verma y J. Pearl. Causal networks: Semantics and expressiveness. En R. D. Shachter, T. S. Levitt, L. N. Kanal y J. F. Lemmer, editores, "Uncertainty in Artificial Intelligence 4", págs. 69–76. North-Holland (1990).

[346] T. Verma y J. Pearl. Equivalence and synthesis of causal models. En "Proceedings of the 6th Annual Conference on Uncertainty in Artificial Intelligence (UAI-91)", págs. 220–227. Elsevier Science (1991).

[347] P. Walley. Inferences from multinomial data; learning about a bag of marbles (with discussion). *Journal of the Royal Statistical Society, Series B* **58**, 3–57 (1996).

[348] F. Wang, D. Pan y J. Ding. A new approach combined fuzzy clustering and Bayesian networks for modeling gene regulatory networks. En "BMEI '08: Proceedings of the 2008 International Conference on Bio-Medical Engineering and Informatics", págs. 29–33. IEEE Computer Society (2008).

[349] M. Wang, Z. Chen y S. Cloutier. A hybrid Bayesian network learning method for constructing gene networks. *Computational Biology and Chemistry* **31**(5-6), 361–372 (2007).

[350] S. M. Weiss y C. A. Kulikowski. "Computer systems that learn: Classification and prediction. Methods from statistics, neural nets, machine learning and expert systems", cap. Chapter 2: How to estimate the True Performance of a Learning System, págs. 17–49. Morgan Kaufmann Publishers Inc. (1991).

[351] A. V. Werhli, M. Grzegorczyk y D. Husmeier. Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical Gaussian models and Bayesian networks. *Bioinformatics* (2006).

[352] A. V. Werhli y D. Husmeier. Reconstructing gene regulatory networks with Bayesian networks by combining expression data with

multiple sources of prior knowledge. *Statistical Applications in Genetics and Molecular Biology* **6**(1), 15 (2007).

[353] A. V. Werhli y D. Husmeier. Reverse engineering gene regulatory networks with Bayesian networks from expression data combined with multiple sources of biological prior knowledge. En "11th Annual International Conference on Research in Computational Molecular Biology (RECOMB 2007)" (2007).

[354] N. Wermuth y S. Lauritzen. Graphical and recursive models for contingency tables. *Biometrika* **72**, 537–552 (1983).

[355] M. West. Bayesian factor regression models in the large p, small n paradigm. En "Bayesian Statistics 7. Proceedings of the Seventh Valencia International Meeting", págs. 723–732. Oxford University Press (2003).

[356] J. Whittaker. "Graphical models in applied mathematical multivariate statistics". John Wiley and Sons (1990).

[357] P. Wilks y M. English. Accurate segmentation of respiration waveforms from infants enabling identification and classification of irregular breathing patterns. *Medical Engineering and Physics* **16**(1), 19–23 (1994).

[358] E. Wit y J. Mcclure. "Statistics for microarrays". John Wiley & Sons (2004).

[359] I. H. Witten y E. Frank. "Data mining: Practical machine learning tools and techniques". Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, second ed. (2005).

[360] D. H. Wolpert y W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* **1**(1), 67–82 (1997).

[361] M. L. Wong, W. Lam y K. S. Leung. A guide to the literature on learning probabilistic networks from data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(2), 1745–178 (1999).

[362] K. S. Woods, C. C. Doss, K. W. Vowyer, J. L. Solka, C. E. Prieve y W. P. J. Kegelmeyer. Comparative evaluation of pattern recognition techniques for detection of microcalcifications in mammography. *International Journal of Pattern Recognition and Artificial Intelligence* **7**(6), 1417–1436 (1993).

[363] G. Wright, B. Tan, A. Rosenwald, E. H. Hurt, A. Wiestner y L. M. Staudt. A gene expression-based method to diagnose clinically distinct subgroups of diffuse large B-cell lymphoma. *Proceedings of National Academy of Sciences of the U S A* **100**(17), 9991–6 (2003).

[364] G. xia Liu, W. Feng, H. Wanga, L. Liu y C. guang Zhou. Reconstruction of gene regulatory networks based on two-stage Bayesian network structure learning algorithm. *Journal of Bionic Engineering* **6**(1), 86–92 (2009).

[365] Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai y T. P. Speed. Normalization for cDNA microarray data: A robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research* **30**(4) (2002).

[366] D. Zak, F. Doyle, G. Gonye y J. Schwaber. Simulation studies for the identification of genetic networks from cDNA array and regulatory activity data. En "Proceedings of the Second International Conference on Systems Biology", págs. 231–238 (2001).

[367] B. T. Zhang y K. B. Hwang. Bayesian network classifiers for gene expression analysis. En D. Berrar, W. Dubitzky y M. Granzow, editores, "A Practical Approach to Microarray Data Analysis", págs. 150–165. Kluwer Academic Publishers (2003).

[368] Y. Zhang, Z. Deng y P. Jia. A new dynamic Bayesian network for integrating multiple data in estimating gene networks. En "ICNC '07: Proceedings of the Third International Conference on Natural Computation", págs. 264–269. IEEE Computer Society (2007).

[369] Y. Zhang, Z. Deng, H. Jiang y P. Jia. Inferring gene regulatory networks from multiple data sources via a dynamic Bayesian network with structural EM. En S. C. Boulakia y V. Tannen, editores, "Data Integration in the Life Sciences, 4th International Workshop,

DILS 2007, Proceedings", vol. 4544 de "Lecture Notes in Computer Science", págs. 204–214. Springer (2007).

[370] X. Zhou, X. Wang y E. R. Dougherty. Construction of genomic networks using mutual-information clustering and reversible-jump Markov-chain-Monte-Carlo predictor design. *Signal Processing* **83**(4), 745–761 (2003).

[371] X. Zhou, X. Wang, R. Pal, I. Ivanov, M. Bittner y E. R. Dougherty. A Bayesian connectivity-based approach to constructing probabilistic gene regulatory networks. *Bioinformatics* **20**(17), 2918–2927 (2004).

[372] J. Zhu, A. Jambhekar, A. Sarver y J. DeRisi. A Bayesian network driven approach to model the transcriptional response to nitric oxide in Saccharomyces cerevisiae. *Public Library of Sciencie (PLoS) ONE* **1**(1), e94 (2006).

[373] J. Zhu, M. C. Wiener, C. Zhang, A. Fridman, E. Minch, P. Y. Lum, J. R. Sachs y E. E. Schadt. Increasing the power to detect causal associations by combining genotypic and expression data in segregating populations. *Public Library of Sciencie (PLoS) Computational Biology* **3**(4), e69 (2007).

[374] J. Zhu, B. Zhang, E. N. Smith, B. Drees, R. B. Brem, L. Kruglyak, R. E. Bumgarner y E. E. Schadt. Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks. *Nature Genetics* **40**, 854 – 861 (2008).

[375] M. Zou y S. D. Conzen. A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics* **21**(1), 71–9 (2005).