PH. D. THESIS

REPRESENTATION OF UNCERTAIN AND IMPRECISE BEHAVIORS AND ITS APPLICATION TO MUSIC PERFORMANCE

Author:
Miguel José MOLINA SOLANA

Supervisors:
Miguel Delgado Calvo-Flores
Waldo Fajardo Contreras

Universidad de Granada

Editor: Editorial de la Universidad de Granada
Autor: Miguel José Molina Solana
D.L.: GR 3101-2012
ISBN: 978-84-9028-239-7

La memoria *"Representation of uncertain and imprecise behaviors and its application to Music Performance"*, que presenta D. Miguel José Molina Solana para optar al grado de doctor, ha sido realizada dentro del Programa Oficial de Posgrado en "Ciencias de la Computación y Tecnología Informática" del departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada, bajo la dirección de los doctores D. Miguel Delgado Calvo-Flores y D. Waldo Fajardo Contreras.

Granada, Marzo de 2012

El doctorando	Los directores	
Fdo: Miguel José Molina Solana	Fdo: Miguel Delgado Calvo-Flores	Fdo:Waldo Fajardo Contreras

A mis padres, Miguel e Isabel

Abstract

Many phenomena in the real world can be understood as behaviors, because they follow some underlying rules and behave in characteristic ways. The computational representation and classification of them is a task of great interest for researchers in behavioral sciences as they are in continuous need of new tools and methods to represent and understand behaviors of growing complexity.

With mobile devices becoming ubiquitous, data series are gaining in importance as a suitable representation for the information coming from their sensors. Data series are defined as an ordered sequence of data at given intervals of an indexing variable (e.g. time). Behaviors can be defined as imprecise and uncertain multivariate data series.

Although the problem of representing imperfect data has been addressed many times in the past, the lack of a general and universal solution obligates to build ad-hoc systems for different problems. For that reason, there is still a need to develop new models to represent such information.

Our proposal, which assumes that some kind of commonality exists among instances of the same behavior in a given domain, represents those imperfect data series as a set of probability distributions. To do so, it first transforms the imperfect observations into qualitative values. Then, it selects a dimension of the behavior and uses it to look for correlations with the rest of dimensions. These correlations are expressed as discrete probability distributions.

The model aims to be general enough to be employed in any domain that contains imprecise and uncertain behaviors, but we concentrated in the particular domain of music in order to validate it. Experiments showed that our representation allows to identify violinists in a dataset of monophonic violin recordings from 23 well-known performers, outperforming comparable alternatives.

Departing from that application, we also studied several aspects of the computational representation of music performances, identified the chorus of songs by means of mining frequent patterns, and proposed a framework for automatic music composition.

Agradecimientos

El presente trabajo no hubiera sido posible sin la confianza inicial, y bastante ciega, que tanto Waldo como Miguel tuvieron en mi persona mucho antes de que yo me planteara siquiera la posibilidad de investigar. Ellos me dieron la oportunidad, y la libertad, de trabajar en temas de mi interés, así como los medios para conseguirlo. Confío en que llegados a este punto, no se hayan arrepentido. Conocerlos como personas y compañeros, y no solo como profesores, ha sido una experiencia gratificante y sorprendente que, en cualquier caso, ha compensado las incertidumbres iniciales de embarcarse en una tesis doctoral.

Económicamente, el desarrollo de esta Tesis Doctoral ha sido financiado por el Ministerio de Educación, bajo el Programa de Formación al Profesorado Universitario (referencia AP2007-02119); y por el Ministerio de Ciencia e Innovación, bajo el Proyecto de Investigación Fundamental TIN2009-14538-C02-01.

Durante el curso de estos años, he tenido la oportunidad de realizar varias estancias en distintos centros de investigación. Sin duda, lo aprendido en ellos, tiene su contribución en esta tesis:

Mi gratitud a Josep Lluis Arcos del *Institut d'Investigació en Intel.ligència Artificial* en Bellaterra, quien me mostró que era posible trabajar conjuntamente en temas tan distantes como Música e Informática, así como a todos los que contribuyeron a que mi estancia en Barcelona y en el *IIIA* fuera como sentirse en casa.

Mi agradecimiento también a todos los que hicieron de mi estancia en Linz una experiencia tan provechosa, especialmente al profesor Gerhard Widmer por corresponder amablemente a mi interés por trabajar en su magnífico grupo, en la universidad Johannes Kepler.

Finalmente, mi agradecimiento al profesor Geraint A. Wiggins y a su grupo del Goldsmiths College en Londres, por su hospitalidad, conversaciones en *The Hobgoblin* e interés en mi trabajo. Y sobretodo, a quienes me ayudaron a disfrutar de la fascinante ciudad de Londres, pese a las dificultades.

Por supuesto, no puedo olvidarme de las personas que durante estos últimos años han estado más cercanas compartiendo despacho, cafés, comidas y tertulias. A todos, ¡sois más de veinte!, gracias. También, como no, de los que siempre han estado cercanos, los amigos de toda la vida.

Dentro del departamento, no me olvido de Miguel García Silvente y de Rosa, quienes en todo momento me han ayudado con la burocracia. Sin ellos, muchas cosas hubieran sido aun más tediosas de lo que ya eran.

Quiero tener un recuerdo también para mi familia, a los que están y a los que estuvieron. Ellos siempre me han animado y elogiado, en ocasiones inmerecidamente. No obstante, si alguien ha creído en mi desde el principio, han sido mis padres. Con su esfuerzo, sacrificio y constancia, casi nunca reconocido, han logrado llevarme hasta aquí, siendo ellos el mejor ejemplo a seguir. Gracias. Mis hermanos, Rocío y Pedro, también han sido durante todos estos años compañía y fuente de orgullo, pese a no haberles dedicado todo el tiempo que se merecen.

Investigar (en España) es un camino arduo y extenuante. Compartir con Virginia los sufrimientos, incertidumbres y alegrías ha sido, es y será un recordatorio de que lo importante en la vida no está, ni mucho menos, en los artículos, las citas y los congresos.

Resumen

En cualquier campo se pueden encontrar fenómenos que, aún compartiendo objetivos comunes, se comportan de forma distinta para alcanzarlos. Dependiendo del punto de vista del observador, dichos fenómenos pueden ser entendidos como similares o completamente divergentes. Estos aspectos hacen posible dividir a dichos fenómenos en distintos subconjuntos de acuerdo a criterios estructurales, lógicos o estéticos.

Debido al incremento en el número de dispositivos móviles y de sus capacidades sensoras, inmensas cantidades de datos están disponibles, haciendo posible la extracción de valioso conocimiento y permitiendo su potencial entendimiento y representación efectiva.

Esta tesis doctoral presenta una nueva propuesta para modelar y representar computacionalmente este concepto de *comportamientos*, utilizando herramientas tanto de Minería de Datos como de Soft Computing. La investigación se basa en trabajos anteriores de nuestro grupo de investigación de la Universidad de Granada sobre representación de patrones repetitivos e identificación de comportamientos.

Comportamientos

¿Cómo podemos definir un *comportamiento*? Intuitivamente, podemos decir que indica la manera en que algo se realiza; es decir, es un conjunto de acciones o características que son particulares de un individuo o fenómeno. Diferentes comportamientos surgen del hecho de que prácticamente cada fenómeno puede ser realizado de varias maneras manteniendo su propósito original. En otras palabras, cualquier función puede ser implementada de diversas formas, no siendo la configuración de sus partes de especial relevancia para lograr su objetivo, pero sí para determinar su *comportamiento*.

La representación ajustada y la gestión de dichos comportamientos es una tarea de indudable interés en las Ciencias de la Computación debido a que los nuevos dispositivos están cada vez más extendidos, y su empleo para tareas de monitorización es muy común.

No en vano, cuando se monitoriza, el objetivo final consiste en identificar el fenómeno subyacente que da forma al comportamiento visible. En ese sentido, podemos entender un *comportamiento* como la huella dactilar de un determinado fenómeno.

No obstante, los *comportamientos* en la vida real rara vez son exactos. Aunque generalmente están compuestos por patrones repetitivos, las repeticiones no son ni idénticas ni predecibles. Es decir, no podemos estar completamente seguros de qué va a pasar a continuación aunque tengamos una buena estimación. Para complicar las cosas, los valores concretos obtenidos mediante sensorización no suelen ser representativos en la mayoría de los casos, donde lo importante es la tendencia general. Por todo ello, podemos decir que los comportamientos reales son imprecisos e inciertos.

Representar *comportamientos* de una manera adecuada puede ser útil para crear nuevos ejemplos artificiales que reflejen la esencia de uno previo. Aunque esto podría considerarse plagio o imitación, no deja de ser un mecanismo muy útil para obtener ejemplos en tareas en las que resulta muy costoso hacerlo. En realidad, podría darse la vuelta al proceso y servir también para comprobar si un fenómeno desconocido se deriva de otro o no.

En cualquier caso, la representación y el resumen de *comportamientos* puede resultar de especial relevancia para entender mejor cómo se comporta un fenómeno, dándole a los expertos nueva información sobre las reglas que lo gobiernan.

Los *comportamientos*, tal y como los hemos definido, pueden encontrarse en numerosos campos. Los siguientes son sólo unos cuantos ejemplos:

Clima El clima en un área determinada es uno de los ejemplos más claros de comportamientos, puesto que es cíclico pero nunca exactamente igual. Y sin embargo, todo el mundo tiene una idea de cómo es el clima en su ciudad (aunque sólo pueda definirlo de forma vaga); de igual manera, todos hemos oído el comentario de "el clima aquí es como el de mi ciudad". Ser capaces de representar con exactitud y predecir el clima en un área es, sin duda, un problema de gran impacto económico.

Lesiones Los movimientos de una articulación (lesionada o no) cuando se hacen determinados ejercicios de rehabilitación comparten determinadas particularidades que los médicos buscan con especial interés. La representación precisa y el entendimiento de las diferencias entre una articulación lesionada y otra sana son de gran importancia para hacer un diagnóstico correcto e incluso descubrir informes fraudulentos.

Finanzas Las interacciones entre los cambios de divisas es otro problema que puede ser representado con el formalismo de los comportamientos en mente. Es bien conocido que el comportamiento de una divisa depende en gran medida de las fluctuaciones de otras divisas. Entender dicha relación y con otros fenómenos puede dar lugar a importantes ventajas económicas.

Energía El perfil de consumo energético de un hogar o una empresa puede ser considerado también como un comportamiento. Si se le representa correctamente, puede permitir la optimización de los procesos de generación y transporte de energía para ajustarse mejor a la demanda, reduciendo los costes asociados, tanto económicos como medioambientales.

Tal y como se ha dicho anteriormente, la representación y gestión de comportamientos es una de las líneas de investigación que el grupo *ARAI* (Razonamiento Aproximado e Inteligencia Artificial) de la Universidad de Granada persigue con mayor interés. Entre otros, se han realizado trabajos en los campos de Ancient Ambient Living (Delgado et al., 2009b), e interpretación musical (Molina-Solana et al., 2010).

Estilo

Un caso particular de *comportamiento* es el *estilo*, en el que existe un autor (generalmente humano) con intenciones. La complejidad de entender y representar el *estilo* es aún mayor que la de sus equivalentes en *comportamientos*, debido a los aspectos estéticos y emotivos implicados.

De manera informal, cuando hablamos del estilo de un artista, nos referimos al conjunto de características únicas de su trabajo, incluso cuando comparte algunas de ellas con otros artistas. Monet y Picasso pueden ser clasificados conjuntamente como 'pintores', pero si los observamos en detalle, la mayoría de la gente podría decir que existen diferencias relevantes entre los trabajos de uno y otro. Cuáles son esas diferencias y cómo explicarlas es otra cuestión que probablemente sólo los expertos pueden responder con precisión, ya que no está claro qué criterios debemos seguir y cuál es su importancia. Para complicar aún más las cosas, el *estilo* implica a menudo emociones en el observador del fenómeno.

Comúnmente, la primera aproximación para tratar con *estilo* consiste en describir cada fenómeno con todo lujo de detalles. Pero incluso si tratamos de conseguir la descripción más detallada, medir el *estilo* no es una tarea fácil ni directa, ya que no están claros ni los criterios ni la importancia relativa que éstos tienen. Es en este punto, cuando las herramientas para tratar con incertidumbre e imprecisión son requeridas y bienvenidas para describir con éxito el *estilo/comportamiento* mediante ordenadores.

A pesar de todas estas dificultades, la investigación en Estilo Computacional está en pleno auge pues los investigadores se han dado cuenta de la importancia y posibilidades de tratar el *estilo* mediante herramientas informáticas. Aunque inicialmente sólo se centraron en Lingüística, estas herramientas pueden ser aplicadas en la actualidad a numerosos problemas de los que se desee estudiar los patrones subyacentes en los procesos de codificación de información de cualquier tipo. Los siguientes son algunos ejemplos de problemas que pueden ser estudiados y resueltos usando este paradigma:

Autoría de textos: El análisis estilístico de un texto es de especial relevancia para detectar usos idiosincrásicos del lenguaje que permitan distinguir unos autores de otros. No en vano, el estilo de un autor es a menudo su huella de identidad. La cantidad de vocabulario empleado, la longitud y complejidad de las oraciones, o el tono (descriptivo, irónico o informal) son sólo algunos aspectos con los que el autor puede modelar a voluntad su mensaje. De hecho, no es extraño escuchar comentarios como "este libro es muy del estilo de Pérez.Reverte".

Traducción automática: Además de la posibilidad obvia de identificar al autor de un texto, el estilo es un aspecto crucial para realizar traducciones automáticas de un idioma a otro. El tiempo de los verbos, y el uso de la voz activa o pasiva (en la lengua inglesa) son sólo dos ejemplos de cómo el estilo puede usarse en este contexto. Más aún, entender la razón por la cual el lenguaje fue usado de una determinada manera puede ser la clave para traducir de forma efectiva un mensaje a otro idioma.

Interpretación musical: Este problema es muy similar al de la autoría de texto. En este caso en lugar de un mensaje textual se utiliza música y el objetivo es identificar las características personales de cada intérprete. Tras representar e identificar, podemos utilizar dicho conocimiento para predecir el intérprete de una nueva pieza o para replicar el estilo de una interpretación particular.

Presentación de resultados: Cuando existen datos disponibles, un buen resumen, presentación y explicación son cruciales para transmitir la información de manera efectiva al destinatario. El *estilo* en esta área puede determinar si los gráficos son preferibles a varias líneas de texto; o qué colores emplear, la verbosidad o el tamaño. Aunque todas las presentaciones persiguen el mismo objetivo (transmitir información), algunas son más adecuadas que otras en función del contexto. Entender y codificar estos aspectos son tareas de gran interés e importancia.

Justificación

Más y más, los smartphones y los nuevos dispositivos recolectan y envían multitud de datos de forma automática a través de sus sensores embebidos. Esta creciente cantidad de información requiere de continuas mejoras en las capacidades de procesamiento, almacenamiento y resumen, siendo el adecuado entendimiento y representación de dichos datos de gran valor científico y económico. Sin embargo, su naturaleza imprecisa e imperfecta hace que sea muy difícil representarlos con las herramientas computacionales clásicas, ya que dichos comportamientos (definidos como secuencias de datos) no se repiten de manera idéntica sino similar.

Por lo tanto, se necesitan nuevas formas de modelar y entender los *comportamientos* especialmente si éstos son inciertos e imprecisos. El objetivo principal de este trabajo doctoral es precisamente proponer un nuevo modelo de representación para comportamientos que resuma sus aspectos más representativos y que tenga en cuenta la imperfección de las observaciones.

Para ello, nos centraremos en comportamiento descritos como series de observaciones con las siguientes tres características:

- Las series de datos contienen uno o más patrones recurrentes. Dichos patrones serán los que trataremos de identificar.
- Si un determinado patrón fue observado en el pasado, volverá eventualmente a suceder de nuevo en el futuro.
- Las series de datos son lo suficientemente largas como para que la información extraída de ellas sea representativa. Cuanto mayor se la longitud de la serie, más ajustada será la representación conseguida.

Debe indicarse, no obstante, que en el presente trabajo sólo nos centraremos en descripción. No se abordará la explicación de ninguna manera, ya que esa es un área mucho más relacionada con Psicología que con Ciencias de la Computación. Es decir, no trataremos las razones por las que un fenómeno se comporta de determinada manera; con respecto al *estilo*, no estamos interesados en las intenciones ni en los objetivos de los creadores.

En cualquier problema de ingeniería, la existencia de un dominio que pueda ser estudiado y generalizado, y en el que las soluciones puedan ser aplicadas y probadas, es de capital importancia. Históricamente se ha demostrado que las soluciones genéricas en Ingeniería son difícilmente alcanzables de una manera abstracta. En esta tesis, nos centramos en música, que es un medio muy interesante para investigar procesos de aprendizaje implícitos por diversos motivos. El principal de ellos es que la música es una estructura de nuestro entorno demasiado compleja para ser aprehendida mediante pensamiento explícito y razonamiento deductivo. Los eventos musicales son por sí mismos de poca importancia, pero sin embargo, una pieza musical es más que una sucesión de sonidos más o menos agradables. El efecto psicológico del sonido viene de las complejas relaciones entre eventos musicales de una pieza dada.

En realidad, las estructuras musicales son generalmente concebidas para no ser explícitamente procesadas. Los compositores ponen todo su empeño en hacer que los oyentes sean sensible a las estructuras que existen dentro de una pieza, pero sin que estos últimos sean conscientes de ello. Por lo tanto, no es raro que la impresión general entre la audiencia sea la incapacidad para describir verbalmente lo que perciben. En muchos casos, las personas están incluso convencidas de que no perciben ninguna estructura subyacente.

Por otro lado, desde que los primeros ordenadores fueron desarrollados, muchas personas han tratado de utilizarlos en tareas musicales. Los proyectos que aplican ordenadores a tareas musicales pueden clasificarse en dos grupos principales: análisis y composición. El primero consiste en extraer información de la propia música (o los datos asociados) con el objetivo de aprender un modelo que describa los ejemplos concretos. La composición, por otro lado, trata de generar nueva música desde reglas. En realidad, se trata de realizar el proceso en la otra dirección: 'de las reglas a la música' en lugar de 'de la música a las reglas'.

Por estas razones y por mi interés personal como pianista, hemos decidido que sea en Música donde aplicar nuestra investigación en aprendizaje y representación de comportamientos. Este hecho no significa que nos limitemos únicamente a los aspectos particulares de este dominio. Todo lo contrario, pues en todos los desarrollos realizados la abstracción ha estado siempre presente, con el objetivo de obtener una propuesta lo suficientemente general como para ser aplicada en otros campos en etapas posteriores.

Contenidos

El presente trabajo doctoral está organizado en torno a cinco bloques que cubren diversos objetivos y problemas abiertos. Cada uno de estos bloques se encuentra respaldado por sus correspondientes publicaciones en revistas de impacto.

Interpretación musical con técnicas computacionales

Los estudios en interpretación musical tienen una especial relevancia en nuestros días ya que ésta es una tarea compleja y aún no muy bien entendida. No en vano, el arte de interpretar música es el resultado de varios años de entrenamiento. Al mismo tiempo, las actuales tecnologías de la información ofrecen la posibilidad de reproducir música especialmente compuesta para ordenadores o almacenada en bases de datos. En esos casos, la música suele sonar tal y como está escrita en la partitura, ignorando el valor de la interpretación en vivo, así como su dificultad y diversidad.

En este bloque se muestran diversas investigaciones en interpretación musical, que varían desde los estudios que persiguen entender la expresividad, hasta los que intentan modelar las interpretaciones desde un punto de vista formal, cuantitativo y predictivo. La investigación en este campo busca poner de manifiesto las herramientas expresivas que tradicionalmente han estado ocultas en la habilidad de los músicos y en su intuición musical. Si estos recursos son formulados de forma explícita, pueden posibilitar la reproducción de archivos musicales con diferentes expresividades.

A través de una revisión de distintos trabajos, hemos identificado los problemas más notables a los que los investigadores deben enfrentarse cuando trabajan en esta área. El principal de ellos es la representación de datos, ya que la obtención de éstos es una tarea muy compleja y a menudo poco fiable. Además, las reglas que gobiernan una interpretación no son exactas ni están bien definidas. Por todo ello, los esfuerzos (los nuestros incluidos) van en la dirección de entender y representar adecuadamente cómo tocar una pieza musical.

A pesar de que somos escépticos en cuanto a que un ordenador pueda reemplazar completamente a un intérprete humano, estamos seguros de que esta tecnología estará disponible para ciertas tareas en un futuro no muy lejano. La existencias de máquinas que puedan modelar la música imitando el estilo de un cierto intérprete dejará pronto de ser un producto de ciencia ficción.

Personalmente, estamos plenamente convencidos de que este es un momento ideal para que las Ciencias de la Computación trabajen en el dominio de la música. Los resultados de tales investigaciones tendrán, sin lugar a dudas, un gran impacto tanto en las Artes como en las Ciencias. No en vano, la música es mucho más que un dominio interesante y de alguna manera exótico; es parte de nuestra esencia humana.

Arquitectura para composición e interpretación musical

Dentro del campo de la música, proponemos una arquitectura para componer e interpretar música de forma automática mediante ordenador. Dado que éste es un reto muy complejo y que implica numerosas tareas, se utiliza una aproximación *bottom-up* para abordarlas de una forma modular, de modo que un problema muy grande pueda ser descompuesto en tareas más pequeñas.

En concreto se propone una arquitectura multiagente de dos niveles que permite, además, que los usuarios puedan usar una interfaz muy sencilla que oculte la complejidad inherente a la composición musical. Dicha interfaz recibe entradas emocionales de los usuarios, por lo que nuestro sistema aborda aspectos de la expresividad y emotividad musical.

Se ha descrito cómo ha sido desarrollado el sistema y de qué manera se representa el conocimiento. Asimismo, hemos construido diversos agentes con distintos roles y objetivos.

Aunque *InMaMuSys* está aún en desarrollo, la experimentación llevada a cabo muestra resultados prometedores que nos animan a continuar desarrollando dentro de esta arquitectura. Llegados a este punto, creemos que la Teoría de Agentes es un formalismo adecuado para modelar la complejidad de los procesos de composición e interpretación musicales.

Minería de motivos musicales

En este trabajo doctoral presentamos también una aplicación de la minería de patrones frecuentes para el descubrimiento de motivos en una pieza musical. Para ello se han transformado ficheros *MusicXML*, que pueden ser obtenidos con facilidad, en secuencias de notas definidas a bajo nivel. Nuestro algoritmo, llamado *SSMiner*, es capaz de identificar de forma eficiente las subsecuencias frecuentes dentro de una secuencia mayor.

En música es bien conocido que los patrones repetidos no tienen por qué ser idénticos. Por ello, nuestro algoritmo es capaz de descubrir los patrones transportados, incluidas las repeticiones exactas (que no son más que transportes nulos). La experimentación realizada indica que nuestro enfoque tiene buenos resultados con un conjunto de canciones seleccionadas aleatoriamente.

Frequent Correlated Trends para representar comportamientos

Como hemos comentado, numerosos fenómenos del mundo real pueden ser entendidos como comportamientos, ya que siguen diversas reglas subyacentes y se realizan de formas particulares. Su representación computacional y su clasificación son tareas de gran interés para los investigadores en ciencias del comportamiento, que continuamente necesitan nuevas herramientas y métodos para representar y entender comportamientos de creciente complejidad.

Las series de datos son una de las representaciones con más importancia en los últimos tiempos. Se las puede definir como una secuencia ordenada de datos a intervalos fijos de una variable. Un comportamiento puede definirse a menudo como una serie multivaluada de datos imprecisos e inciertos.

Aunque el problema de la representación de información imprecisa se ha abordado numerosas veces en el pasado, la falta de una solución general obliga a construir sistemas ad-hoc para las diferentes aplicaciones que se presentan. Por esta razón, sigue siendo necesario encontrar nuevas soluciones y modelos para representar este tipo de conocimiento.

Nuestra propuesta, que asume la existencia de aspectos comunes entre varias instancias de un mismo comportamiento, representa dichas series de datos imperfectas como un conjunto de distribuciones de probabilidad. Para ello, primero se transforman las observaciones imperfectas en una secuencia de valores cualitativos. A continuación, se selecciona una dimensión del comportamiento y se buscan las correlaciones de ésta con el resto de dimensiones, que se expresarán como distribuciones de probabilidad.

La principal ventaja de nuestro modelo, es que emplea una representación para comportamientos finita y constante en tamaño, sin importar el número de observaciones disponibles, aunque consigue sus mejores resultados conforme aumenta el número de observaciones. Permite asimismo una representación incremental hasta un instante determinado.

Frequent Correlated Trends en interpretaciones musicales

La viabilidad del modelo anteriormente descrito se ha probado mediante su aplicación al dominio de las interpretaciones musicales. En particular, nos hemos centrado en la tarea de identificar al intérprete de una pieza musical por su forma de tocar, haciendo uso de las *frequent correlated trends* (tendencias frecuentes correladas) para capturar sus tendencias expresivas. La forma de tocar un instrumento puede considerarse un comportamiento, tal y como lo hemos previamente definido.

El proceso de aprendizaje parte de herramientas de extracción de audio y de una segmentación automática de las melodías. A los intérpretes se los representa mediante un conjunto de distribuciones de probabilidad que capturan su estilo personal con respecto a una colección de patrones melódicos. En nuestra investigación mostramos que, sin una gran precisión en la obtención de datos, nuestra propuesta es bastante robusta para representar las pequeñas desviaciones de los distintos intérpretes.

La experimentación realizada se ha concentrado en identificar al violinista que toca una pieza, haciendo uso de las duraciones y del volumen de las notas en sus interpretaciones. Se han utilizado grabaciones de 23 violinistas profesionales diferentes. Los resultados indican que nuestro modelo de *tendencias frecuentes correladas* es capaz de identificar patrones interpretativos que son útiles para distinguir a unos intérpretes de otros. Los resultados, por otro lado, claramente mejoran a los de un clasificador aleatorio, y son, con seguridad, bastante difíciles de conseguir por un oyente humano.

Contribuciones

Las contribuciones más relevantes de la presente tesis doctoral con respecto a los contenidos anteriormente descritos son las siguientes:

Entender la interpretación musical. Debido a que la interpretación musical es el problema particular en el que vamos a aplicar y probar nuestra propuesta de representación de comportamientos, pretendemos estudiar las características propias de este dominio, así como las tareas más relevantes y los trabajos realizados. Por ello, se ha llevado a cabo una revisión de las diferentes alternativas propuestas en la literatura en cuanto a técnicas computacionales aplicadas a la interpretación musical.

Aplicación del DataMining al dominio musical. Como parte de nuestro estudio en herramientas computacionales aplicadas al dominio de la música, hemos desarrollado un algoritmo para descubrir patrones frecuentes dentro de una canción. En concreto, nos hemos centrado en la tarea de identificar el estribillo de diferentes canciones como una manera de estudiar cómo los patrones son repetidos en este contexto.

Desarrollo de una arquitectura de trabajo. Con el objetivo de integrar todas las herramientas de aprendizaje, se ha propuesto un marco de trabajo y una arquitectura para la composición e interpretación musical automática utilizando ordenadores.

Representación de comportamientos imprecisos e inciertos. Tal y como se ha indicado con anterioridad, nuestra meta principal es la de representar comportamientos imprecisos e inciertos, ya que éste es un tema de relevancia en la actualidad. Tras un estudio de las distintas alternativas y enfoques existentes, hemos estudiado y descrito una representación abstracta para acomodar fácilmente diferentes comportamientos solucionando algunas de las dificultades actuales.

Aplicación a la representación de interpretaciones musicales. Finalmente, se ha aplicado el modelo abstracto desarrollado en el objetivo anterior a un dominio concreto con la intención de probar su validez y rendimiento. Este dominio es el de las interpretaciones musicales, que es un tema que suscita varios problemas de interés. En particular, nos hemos centrado en la tarea de identificar el intérprete de una grabación musical haciendo uso de sus similitudes con interpretaciones anteriores.

Índice general

Parte I.

PhD Dissertation

1. Introduction

> *"What is it that distinguishes a Monet from a Picasso? A Mozart concerto from a Bach fugue? Ballet from hip-hop? Melville from Dickens? Jazz from bluegrass? Valley-speak from the Queen's English?*
> Style"

1.1. Motivation

In any field, we can find phenomena that despite sharing common objectives, behave in dissimilar ways in order to achieve them. Depending on the observer's point of view, those phenomena can be understood as similar or completely divergent. Those differing aspects make it possible to divide similar phenomena in different subsets according to structural, logical or aesthetic criteria.

With the increase of mobile devices and their sensing capabilities, huge amounts of data are becoming available, making it possible to extract valuable knowledge from them, and potentially allowing their understanding and effective representation.

This dissertation presents a novel proposal for computationally modeling and representing this concept of *behaviors*, using tools from both the Data Mining and Soft Computing fields. This research builds upon former works in the representation of repeating patterns and the identification of behaviors from our research group at the University of Granada.

Behaviors

But how can we define *behaviors*? Intuitively, it can be simply said that *behaviors* is about the manner in which something is done or made, regardless of its function; that is, a set of actions or features that are characteristic of one particular individual or phenomenon. Different *behaviors* arise from the fact that almost every phenomenon can be performed in different ways while still serving its original purpose. In other words, any given function can be implemented in several ways, and the actual configuration of its component parts is of no importance to its function, but dramatically shapes the *behavior*.

The accurate representation and management of these *behaviors* is hence a task of unquestionable interest in Computer Sciences as new computing devices are becoming ubiquitous, and monitoring is becoming a widespread task. Not in vain, when monitoring, the final aim usually consists in identifying the underlying phenomenon that rules the actions: the *behavior*. In that sense, we can understand *behaviors* as the signature of a given phenomena.

However, *behaviors* in real life are everything but exact. They are composed of repeating patterns, but those repetitions are not identical nor fully predictable. In other words, we can never be completely sure of what is going to happen next despite we might have an educated guess. Furthermore, the exact measures are not representative in most cases, being the overall trend much more valuable. Because of that, real behaviors are said to be imprecise and uncertain.

Successful representation of *behaviors* can be used for creating new artifacts that reflect the essence of a previous one. Although this action might be considered as plagiarism or imitation, it is helpful for tasks in which obtaining new examples is costly. What is more, we can invert that process to check whether an unknown phenomenon is derived from any other or not.

In any case, representation and summarization of *behaviors* can provide a valuable understanding of how a phenomenon behaves, giving experts new pieces of information about the rules that govern the behavior of that particular phenomenon.

Behaviors, as previously defined, can be found in many fields. The following are a few examples of them:

Weather The weather in a given area is one of the clearest examples of behaviors. It is cyclical but never exactly the same. However, everyone has an idea of how the weather in his city is (even though we can only define it in vague terms), and we all hear comments like "the weather here is like in my city". Being able to accurate represent and predict the weather in an area is a problem of great economic impact.

Injuries The movements of a (injured) knee when doing some rehabilitation exercises share common nuances that doctors look for with special interest. Accurately representing and understanding the differences between an injured articulation and a healthy one is of great help to correctly diagnose the injury and to discover fake reports.

Finances The interactions between currency exchanges is also a problem that can be represented with the formalism of behaviors in mind. It is well known by experts that a currency's behavior is in great degree dependent of other currencies' fluctuations. Understand the relationship between them and with other phenomena might lead to important economic advantages.

Energy The energy consumption profile of a family or a business is also a behavior. If accurately represented, it would allow potential optimization in the generation and transport processes to better match the demand, reducing the associated costs, both economic and environmental.

As said before, the management and representation of behaviors is one of the lines of research the *ARAI* (Approximate Reasoning and Artificial Intelligence) group at the University of Granada is pursuing with greatest efforts. Among others, research have been done in the fields of Ancient Ambient Living (Delgado et al., 2009b), and music performance (Molina-Solana et al., 2010).

Style

One particular case of *behaviors* is *style*, in which an author (generally a human) with intentions exists. The complexity of understanding and representing *style* is even bigger than their counterparts with *behaviors*, due to the aesthetics aspects and emotions involved.

Informally, when talking about an artist's style, we are referring to a set of characteristics unique to the work of that individual artist, even when sharing some of them with other artists[1]. Monet and Picasso might be classify together as 'painters', but if we see them in-depth, most people can intuitively tell that there are some differences between the drawings they produced. Which differences those are and how to explain them is another question that probably only experts can accurately answer, as it is not clear which criteria

[1] The cite at the beginning of this Chapter, extracted from *The Structure of Style* (Argamon et al., 2010), summarizes this idea.

we must attend to and how important they are. To complicate things more, *style* often implies emotions that are recalled on the observer of the phenomenon.

Typically, the very first approach to tackle *style* consists in describing each phenomenon in every detail. But even if we try our best to achieve the most detailed of the descriptions, measuring *style* is not an easy or straightforward task, as it is not clear which criteria we must attend to and how important they are. It is at this point when tools for dealing with uncertainty and imprecision are required should we want to successfully describe and manage *style/behaviors* with computers.

Despite those difficulties, Computational Stylistics is a growing research field as more people realize the importance of addressing *style* by means of computational tools. While initially only focused on Linguistic, those tools can now be applied to any other problem in which we want to study the patterns underlying in the process of encoding information of any type. The following are some examples of problems that can be studied and solved using this paradigm:

Text authoring: Stylistic analysis of a text is key to the detection of idiosyncratic uses of language that distinguish one author from another. Not in vain, an author's style is often his signature. The range of vocabulary, the length and complexity of sentences, or the tone (descriptive, ironical, informal) are just a few aspects which the author can shape their message with. In fact, it is not strange to hear comments like "this book is much in the style of Dan Brown".

Machine Translation: Besides that obvious possibility of identify the author of a text, style is a crucial aspect when automatic translating from a language to another. The tense of verbs, or the use of active/passive voice are only two examples or how style might be used in this context. Even more, understanding why a language was used in a certain way might be key to translate the message effectively to another language.

Music performing: This problem is very similar to the one of text authoring. In this case, instead of a textual message it is music what we have, and the aim is to identify each performer's own signature. After that, we can apply such knowledge to the task of predicting the performer of a new piece, and to the task of replicating the style of a particular performance.

Results presentation: When some data are available, a good presentation, summarization and explanation are key if we want to effectively transmit the information to users. *Style* in this area might determine if graphs are preferred to a few lines of text. It also may determine the colors, verbosity or size. Even though all presentations pursue the same goal, some are more appropriate and effective regarding the context. Understanding and encoding those aspects is of great importance and interest.

1.2. Justification

More and more, smartphones and new devices automatically collect and send many kinds of data from multiple domains through their embedded sensors. This growing amount of data is continuously demanding new processing, storage and summarization capabilities, and their understanding and representation is of great scientific and economic value. However, their imprecise and imperfect nature make them very difficult to be represented with classical computational tools, as those behaviors (defined as sequences of data) do not recur in an identical but similar way.

Therefore, new ways of modeling and understanding *behaviors* are needed specially if they are uncertain and imprecise. The main aim of this dissertation is precisely proposing a new representation model for behaviors summarizing their most representative aspects taking into account the aforementioned imperfection of observations.

In this thesis, we will concentrate on behaviors described as series of observations with the following three characteristics:

- The series of data contain one or several recurrent patterns. Those patterns will be the ones we will try to identify.
- If a certain pattern was observed in the past, it will eventually happen again in the future.
- The data series are long enough for the information obtained from them to be representative. The longer the data series is, the more accurate the representation.

It should be noted, however, that this dissertation only deals with description. We will not treat explanation in any way, as that is a huge area much more related with Psychology than with Computer Science. In other words, we will not go into the reasons why a particular phenomenon is behaving in a particular way. Referring to *style*, we are not interested here in creators' intentions and aims.

In any engineering problem, the existence of a domain that can be studied, abstracted, and in which solutions can be applied and tested, is of capital importance. Historically, it has been demonstrated that general solutions in Engineering are difficult to be devised in abstract. In this dissertation, we concentrate on music, which is an interesting medium to investigate implicit learning processes for several reasons. The main one is that music is a structure of our environment that is too complex to be apprehended through explicit thoughts and deductive reasoning. Musical events per se are of no importance, but musical pieces are more than a pleasing succession of sounds. The psychological effects of musical sounds come from the complex relationships between musical events in a given piece.

In fact, musical structures are generally not conceived for being explicitly processed. Composers do their best to make listeners sensitive to the structures that underlie a musical piece but unaware of them. Therefore, it is not rare that the most common impression among a general audience is that of being unable to verbally describe what they perceive. In many cases, people are even convinced that they do not perceive any underlying structure.

Since first computers were developed many people have tried to apply them to musical tasks. There are two main classes in which computer music projects could be classified: analysis and composition. The first one consists in extracting information from the music itself (or the associated data) in order to learn a model that describes the concrete examples. Composition, on the other hand, is about generating new music from the rules. In fact, is doing the process in the other way: 'from rules to music' instead of 'from music to rules'.

Because of those reasons and my personal interest as a pianist, Music was chosen as the domain to which apply our research on learning and representing behaviors. This fact does not mean that we limited ourselves to only consider particular aspects of this domain. More on the contrary, abstraction has always been present in our research, with the aim of our results being general enough to be applied to as many areas as possible in later stages.

1.3. Objectives

The present dissertation is organized around the following five objectives that cover several issues and open problems:

Understanding music performances. Because music performance is the particular domain in which we will be applying our behavior representation model, we aim to study the particularities of this domain, and the relevant tasks and former research in it. One of the goals of this dissertation is therefore to provide a survey of the state of the art in computational techniques applied to music performances.

Application of DataMining to the music domain. As part of our study on computational tools applied to the music domain, we aim to develop an algorithm for discovering frequent patterns in compositions. In particular, we focused on the task of identifying the chorus of different songs, as a way to study how patterns are repeated in this context.

Development of a framework. With the aim of integrating all the learning tools, one of the objectives of the present dissertation is the proposal of a framework for automatically composing and performing music by means of computers. This framework should enable the inclusion of the learning algorithms devised in other objectives.

Representation of imprecise and uncertain behaviors. As said before, our primary goal is to represent imprecise and uncertain behaviors, as it is an issue of current interest. After an study of existing alternatives and approaches, we aim to develop an abstract representation that can easily accommodate different behaviors overcoming current difficulties.

Application to represent music performances. Finally, we aim to apply the abstract model in the previous objective to a concrete domain in order to test its validity and performance. This domain is the one of music performances, as it is an issue that poses several interesting problems. We will focus on the task of identifying the performer of several music recordings by its similarity to previous performances.

1.4. Thesis outline

This thesis is divided in two main parts:

PhD Dissertation is devoted to describe the problems we have addressed and discuss the research we have performed. In particular, Section 2 presents the research and the main results, and Section 3 summarizes them and also points to further work.

Publications collects the journal papers related with the research shown in this thesis. They are devoted to the five objectives previously mentioned.

2. Discussion of Results

This chapter is devoted to present the research that has been carried out in several areas and to highlight its most relevant points. We have organized this chapter in several sections corresponding to the different objectives we aimed to cover in this dissertation. We first introduce the particular domain of computational music performance and concentrate on one to-be-solved problem: representing and describing music performances. Later, we abstract this problem to the one of representing imprecise and uncertain behaviors by means of repetitive patterns, and propose a suitable representation model and a learning algorithm. Finally, we test our proposal in the studied domain in order to check its validity and performance.

2.1. Understanding music performances

Most people would judge the literal execution of a musical score to be significantly less interesting than a performance of that piece by even a moderately skilled musician. Why is that so? Because what we hear is not a literal rendition of the score. Of course, the principal and traditional vehicle of communicating musical compositions is the music score, in which the composer codifies his intentions; but the information written in the score does not represent an exhaustive description of the composer's aims. Although it carries information such as the rhythmical and melodic structure of the piece, it is not a notation capable of accurately describing the timing and timbre characteristics of the sound.

In the same way that there is no explicit notation in a written poem for pronunciation, in musical scores there is also such a lack of information. When speaking, we use several voice resources such as changing velocity, tone or loudness. All this effects are not explicitly in the text we are reading. In fact, when several people read a text, resulting sounds are not the same, even though words in the paper remain unchanged. So does in music. This comparison is actually quite appropriate because former research on music performance has revealed interesting analogies in the communication of emotions in singing and speech (Bresin and Friberg, 2000; Sundberg, 2000).

Due to that freedom and lack of precision, the role of the performer is crucial in music, acting as a kind of mediator between composer and listener, between written score and musical sound. It is the performer who renders each note in the score in terms of intensity, duration and timbre by movements of fingers, arms or mouth. This results in different performances of the same piece reflecting each performer's culture, mood, skill and intention. These variances also contribute to determining the performing styles of different musicians. So that, the music we hear has two main sources: the score and the performance, and they both need from each other.

Widmer and Goebl (2004) defined *expressive music performance* as "the deliberate shaping of the music by the performer, in the moment of playing, by means of continuous variations of parameters such as timing, loudness or articulation". But besides performers' intentions, there are other uncontrolled factors affecting the rendition of a musical piece.

One of the most obvious is the physical condition of the performer. Not in vain, performer's mood, health and fatigue play a crucial role in the process of playing an instrument. Some studies (Gabrielsson, 1995; Rigg, 1964) have shown major variations in renditions by the same performer when he is in different moods.

Manual abilities are also an important point that is especially visible when comparing a beginner with an expert. With practice, musicians can improve their velocity and precision, reducing the amount of unintended deviations with respect to the score (commonly known as errors). Other factors that affect the rendition are the location where it takes place and the instrument being used: the acoustics of the place are important because they establish the sounds that can be made; and so does the instrument, which has an evident influence on the character of the work.

Computational research on music performance

As seen, many aspects can affect musical renditions and they are very difficult to be explicitly described. Even more, the concept of a creative activity being predictable and the notion of a direct 'quasi-causal' relation between the musical score and a performance are both problematic.

Despite those difficulties, several authors have attempted to capture common performance principles by means of focusing on commonalities between performances and performers, to later represent them in computers (see reviews from Poli (2004); Widmer and Goebl (2004)).

Two goals are mainly aimed in those systems: automatic style replication, and learning about the artistic activity of expressive music performance. In either case, it is not enough with just extracting information from performance measurements and use it to compare and classify; in order to get real insight, learning algorithms that produce interpretable models are needed.

We reflected on those questions and reviewed several of those attempts and computational models in:

Delgado, M., Fajardo, W. & Molina-Solana, M. (2011), "*A state of the art on computational music performance*", Expert Systems with Applications. Vol. 38(1), pp. 155–160, DOI: 10.1016/j.eswa.2010.06.033

For instance, Juslin et al. (2002) described the main sources of expressivity in musical renditions and expressed the necessity of integrating some of this aspects in a common model they started to sketch.

López de Mántaras and Arcos (2002) studied the expressivity of several AI-based systems for music composition. They compared this expressivity with the one that exists in human recordings. Moreover, they introduced *SaxEx*, a system capable of generating expressive performances of jazz ballads by using examples from human performers and a case-based reasoner.

Hong (2003), on the other hand, studied how musical expressivity is affected by tempo and dynamics variations. He employed cello recordings for the experiments. He extended previous work by Todd (1992), by applying new musical ideas to Todd's model.

Dovey (1995) proposed an attempt to use inductive logic in order to determine the rules that pianist Sergei Rachmaninoff may have used in their performances with an augmented piano. The aim was to extract general rules (in the form of universal predicates) about each note's duration, tempo and pressure. All that information was obtained from the way of playing the piano.

The group led by Gerhard Widmer has worked in the automatic identification of pianists. In (Widmer et al., 2003) they studied how to measure several aspects of performances by applying machine learning techniques; whereas in another work (Stamatatos and

Widmer, 2005), they proposed a set of simple features that could serve to represent performer's expressivity from a rendered musical piece.

Moreover, in a later paper, Saunders et al. (2008) represent musical performances as string of symbols from an alphabet. Those symbols contain information about changes in timing and energy within the song. After that, they use *Support Vector Machines* to identify the performer in new recordings.

Sapp (2007) is also an interesting proposal, as it represents musical renditions by means of sketches which are based on the correlation between time and energy.

Automatic music performance

As said, one of the ultimate goals —and probably the most appealing— of researching in computational music performance is applying the gathered knowledge to the task of automatic performing musical pieces. Because the conventional score is quite inadequate to describe the complexity of a musical performance, and since the literal synthesis of notes from a score is flat and unappealing, there is an opportunity for learning systems that can automatically produce compelling expressive variations. Hence, methods for automatically "bringing life" to musical scores become useful and interesting.

The principal characteristic of an automatic performance system is that it converts a music score into an expressive musical performance typically including time, sound and timbre deviations from a deadpan realization of the score. Mostly, two strategies have been used for the design of performance systems, the analysis-by-synthesis method and the analysis-by-measurement method.

The first method implies that the intuitive, nonverbal knowledge and the experience of an expert musician are translated into performance rules. These rules explicitly describe musically relevant factors. This method is potentially limited because of rules mainly reflecting the musical ideas of specific expert musicians. On the other hand, professional musicians' expertise possess a certain generality, and rules produced with the analysis-by-synthesis method have often been found to have a general character. Those rules failed to apprise the small nuances that performers are unaware of.

Rules based on an analysis-by-measurement method are derived from measurements of real performances usually recorded on audio CDs or played with MIDI-enabled instruments connected to a computer. Data are processed statistically so that the rules reflect typical, rather than individual, deviations from a deadpan performance, even though individual deviations may be musically highly relevant.

Besides most works already cited in the former paragraphs, other authors have proposed models of automatic music performance. Todd (1992) presented a model of musical expression based on an analysis-by-measurement method. Rule-based systems have been proposed by Zanon and Poli (2003) and by Friberg (1991); Friberg et al. (2000).

Performance systems that learn by means of machine learning techniques have been developed too. Widmer (2003) proposed a machine learning based system extracting rules from performances. Ishikawa et al. (2000) developed a system for the performance of classical tonal music; a number of performance rules were extracted from recorded performances by using a multiple regression analysis algorithm. Arcos et al. (1998) developed a case-based reasoning system for the synthesis of expressive musical performances of sampled instruments.

Although it may sound odd, there are concrete attempts at elaborating computational models of expressive performance to a level of complexity where they are able to compete with human performers. The Rendering Contest (Rencon) [1] (Hashida et al., 2011) is an annual event first launched in 2002. It tries to bring together scientist from all over the world for a competition of artificially created performances. It uses an human judge to evaluate music performances automatically generated by computers. Participants are asked to generate a rendition of a musical work by using a predictive level. In a wider sense, we can somehow see this paradigm as an expressive performance *Turing test*. In other words, the best systems are those than manage to generate performances which sounds indistinguishable from human ones.

All things considered, music performance is an interesting research topic which enables the study of human's emotions, intelligence and creativity. These are precisely the issues Minsky (1992) referred to when he wrote about music as a human activity.

2.2. A framework for computer music composition

On the context of our research on computational tools applied to music, we developed a framework, named *InMaMuSys*, for automatically composing and performing music. This framework was based on the analysis-by-synthesis approach we have already

[1] http://www.renconmusic.org

mentioned. In particular, we proposed a multiagent system to organize the different tasks involved in the generation and performance of music by a computer. It has been published in:

Delgado, M., Fajardo, W. & Molina-Solana, M. (2009), "*INMAMUSYS: Intelligent Multiagent Music System*", Expert Systems with Applications. Vol. 36(3), pp. 4574–4580, DOI: 10.1016/j.eswa.2008.05.028

Many attempts have been made on applying computer to music composition, most of them based on some kind of rule-based system (Friberg, 1991; Lerdahl and Jackendoff, 1983). Genetics has also been applied to composing music, as this is a problem with difficulty in defining the solution process within a huge solution space (Marques et al., 2000; Miranda, 2004).

However, stylistics considerations have been often neglected because of their complexity. It has not been until recently (with the help of advances in computational music performance) that prototypes are aiming expressivity in their compositions (López de Mántaras and Arcos, 2002; Zhang and Miranda, 2006).

Our attempt, *InMaMuSys* has a two-layer multiagent architecture. As a one-for-all agent makes little sense in this complex context, a collection of simple agents specialized in particular tasks is proposed. In the first level —the competitive one—, agents (called *composers*) compete among themselves to be the one chosen for composing. This layer allows an initial separation between compositional styles. Each *composer* announces its abilities, and the system, according with user inputs, selects the composer that better fits.

Composers choose, between other parameters, rhythm, number of voices, and instruments to be used in the compositions. However, they only act as directors, asking others agents from the second level for their collaboration in order to get a composition. This second level contains auxiliary agents that collaborate between them, so this layer should be understood as a collaborative level. We call these agents *voice generators*.

InMaMuSys' initial prototype was developed with several agents in order to generate music in different ways: from just a random composer to more elaborate ones where aesthetic principles were mainly searched.

One of the goals of *InMaMusys* was to hide the underlying complexity and expose only a simple interface that anyone could use by means of indicating which kind of music they want, in terms of emotions. In other words, we aimed to integrate into the system itself the knowledge about how to use tonality, rhythm and instruments in order to get, for instance, sad music.

Taking into account that the evaluation of any musical work is a complex task and often comes down to individual subjective opinion, it is hard to empirically evaluate music compositions, and therefore it is difficult to evaluate the effectiveness of a computer music composition system (Pearce et al., 2002). Because of that, we carried out some preliminary experiments over *InMaMuSys* with the aim of checking whether the system was able to compose music that successfully matches user emotional requests. Results showed that different emotions were successfully identified by listeners, and therefore suggested that the approach was valid.

A well-balanced system was developed because there was a great variety between different executions, and *InMaMuSys* managed to make compositions sound as a whole, and not like several pieces stuck together. Even though the whole music space for a emotion (e.g. sadness) is not completely covered, almost all the system compositions are classified (by humans) under the right emotion.

The development of this framework allowed us to reflect on the difficulties of modeling a complex task such as music composition. It also pointed out the question of which alternative is better and more scalable when trying to obtain the necessary knowledge. On the one hand, we can rely on human expertise and represent their knowledge; on the other, we can employ machine learning techniques and leave the computer find out what is relevant and what is not. So far, *InMaMuSys* heavily counts on the first one, but its improvement necessarily should go on the machine learning direction.

2.3. Mining musical patterns

As said in the Introduction, several structures underlie musical pieces even though they are mostly ignored by listeners. In the context of musicology, the discovery of frequent (musical) patterns is a relevant problem, as several repeating entities such as notes, intervals, rhythms, and harmonic progressions can be found. In other words, music might be seen as a string of musical entities on which pattern recognition techniques can be applied. We studied in particular the automatic extraction of relevant patterns from music pieces in order to identify their chorus. This research has been published in:

Jiménez, A., Molina-Solana, M., Berzal, F. & Fajardo, W. (2011), "*Mining transposed motifs in music*", Journal of Intelligent Information Systems. Vol. 36(1), pp. 99–115, DOI: 10.1007/s10844-010-0122-7

A music motif is the smallest meaningful melody element; as a rule, it is a group of notes no longer than one measure. In terms of human speech, a motif would be like a word. In the same way that sentences consist of words, motifs form musical phrases. A melody is formed by several motifs, which are repeated, developed, and opposed one against another within the melody evolution.

Pattern processing techniques have been applied to musical strings and, despite it is almost impossible to be exhaustive in analyzing the state of the art in musical pattern identification, the work by Cambouropoulos et al. (2001) is a good departing point. Those algorithms can be divided into those that deal with audio signals (using signal processing methods) (Aucouturier and Sandler, 2002; Paulus and Klapuri, 2009; Levy and Sandler, 2008), and those that use symbolic representations (using text mining methods) (Meredith et al., 2002; Rolland, 1999; Pienimäki, 2002).

According to the above considerations, we developed a TreeMiner-based (Zaki, 2005) algorithm to discover frequent melodic and rhythmical patterns in music files, using a symbolic representation approach. The algorithm works with a sequence of notes obtained from a *MusicXML* representation of the song. These notes are defined at their lowest level (i.e., pitch and duration) and in an absolute, not relative, way. In those circumstances, our algorithm is able to identify sequences even when they are transposed. It can be used to find common motifs in several songs and also find repetitions within a song. In particular, we applied this algorithm to the task of finding the longest motifs that are repeated within a single song. Our hypothesis is that those patterns probably correspond to the chorus or the most significant part of the song.

Our proposal: SSMiner

Our algorithm, called SSMiner (Similar Sequence Miner), is based on the POTMiner (Jiménez et al., 2010) frequent tree pattern mining algorithm, a TreeMiner-like algorithm for discovering frequent patterns in trees (Zaki, 2005). POTMiner and its antecessor follow the Apriori (Agrawal and Srikant, 1994) iterative pattern mining strategy, where each iteration is broken up into two distinct phases:

- *Candidate Generation*: A candidate is a potentially frequent subsequence. In Apriori-like algorithms, candidates are generated from the frequent patterns discovered in the previous iteration. Most Apriori-like algorithms, including ours, generate candidates of size $k + 1$ by merging two patterns of size k having $k - 1$ elements in common.
- *Support Counting*: Given the set of potentially frequent candidates, this phase consists of determining their actual support (number of occurrences) and keeping only those candidates whose support is above the predefined minimum support threshold (i.e., those candidates that are actually frequent).

The sequence of a song is scanned twice by our algorithm, in the process of obtaining the frequent elements of size 1. The first scan is needed to save the occurrences of each note and the second one is employed to detect the transposed occurrences of each note. Then, the infrequent notes are pruned and we are ready to apply the two phases of the SSMiner algorithm without checking the original sequence any more.

The execution time of our algorithm is proportional to the number of patterns than can be identified, and quadratic with respect to the size of the sequences.

Our algorithm returns all the frequent patterns of the maximum size indicated by the user (or smaller ones if there are no patterns of such size). As musical motifs are generally no longer than a measure, a value of ten is typically used by default. Nevertheless, this limit can be easily modified since our algorithm can return all the frequent patterns that exist in the song regardless of their size. The resulting output will be the set of frequent patterns that represent the song. The algorithm also returns the positions of the different occurrences of the patterns within the song (including transposed occurrences if needed).

In order to evaluate the goodness of our proposal, we used a corpus of 44 songs from a wide variety of authors, testing whether or not the discovered frequent patterns belong to the chorus of the song —in our experiments, 6.82% of the songs do not have a clear chorus. Table 2.1 shows the percentage of songs which have at least one identified pattern within their chorus. As can be seen, above 60% of the songs fulfill this requirement. Also, it is remarkable that not considering the rhythm results in more patterns belonging to the chorus of the songs. This fact indicates that patterns are not always exactly repeated as themselves, but slightly modified. Although the chorus-belonging criterion appears to be a valid and obvious one, it should be noted that some songs are better identified by patterns which do not belong to the chorus.

	pitch-duration	pitch	transposition-duration	transposition
%Yes	63.64	68.18	63.64	72.73
%No	29.55	25.00	29.55	20.45
%without chorus	6.82	6.82	6.82	6.82

Table 2.1.: Percentage of songs that include at least one identified pattern within their chorus. Each column indicates the different setups we have tested: 1) repetitions with exact pitch and duration, 2) repetitions with exact pitch, 3) transposed repetitions with exact figuration, and 4) transposed repetitions with varying durations.

Apart from its obvious use in musicology, one of the most relevant application in the music domain related to motif extraction is Audio-thumbnailing (i.e., summarizing or abstracting) (Zhang and Samadani, 2007). It provides the user with a brief excerpt of a song that, ideally, contains the main features of the work. Before hearing or purchasing a whole song, it is useful to hear a representative thumbnail of the whole work. This technique is also important for indexing large datasets of songs, which can be browsed more quickly and searched more efficiently if indexed by those small patterns instead of being indexed by the whole song.

2.4. Understanding Behaviors

In recent years, behavioral sciences have received a lot of attention from the informatics perspective. This fact is mainly due to current demands for behavior analysis and understanding outstripping the capability of traditional methods and techniques in behavioral sciences. New computational tools for representing and working with behaviors are very welcomed, and a growing field of research, namely *Behavior Informatics* (Cao, 2010), is receiving increasing recognition.

Intuitively, we can define a behavior as a set of actions that are characteristic of one individual or phenomenon. These actions are ordered (or partially ordered) in some way, and indexed by a variable, which is generally *time*. By representing behaviors, two goals are aimed: identification and tagging of the behaviors, and forecasting future actions within them.

Examples of behaviors

Behaviors, as we have just defined them, can be found in many different domains. The following are some illustrative examples of such phenomena that can be represented and individually identified:

- The weather in a given area, represented as a series of observations at different time instants, including information such as temperature, precipitations or wind speed (Kriegler and Held, 2005).
- The movements of a (injured) knee when doing some rehabilitation exercises, by monitoring the position of several reference points at different time instants (Alonso et al., 2002).
- The way of playing an instrument. A particular performance of a piece of music can be represented as a series of notes with its respective length and energy, among others attributes (Molina-Solana et al., 2010).
- Modeling and inferring human behaviors in Ambient Assisted Living, with the aim of identifying strange actions and situations of potential danger (Delgado et al., 2009b).
- The personalization of mobile services. As mobile devices increase their capacity, new services and applications are developed which need modeling the user behavior and context (Bao et al., 2012).
- The interactions between currency exchanges. Several works have studied how several currencies behave against each other at different financial situations (Zhang and Wan, 2007).

Data series for representing Behaviors

As seen, most of these real-world phenomena can be naturally represented by data series. As databases from most of industrial and biological areas often contain timestamped or ordered records, data series are gaining weight as a suitable source of information, and working with them has become an important machine learning task. Those records are generally obtained in an automatic manner from different sensors.

Two main goals of data series analysis are historically found in literature (Brockwell and Davis, 1991): forecasting and modeling. The aim of forecasting is to accurately predict the next values of the series, whereas modeling aims to describe the whole series. Even though they can be sometimes related, they usually differ as finding a proper model for the long-term evolution might not be the best approach to predicting the short-term evolution and viceversa. Forecasting and modeling are also the main tasks concerning behaviors. Therefore, data series are a suitable representation for behaviors, being also the most common one.

Imperfect data series

In either case, and whatever the goal of a particular data series analysis is, data representation is a crucial task anyway. It is hence required a formal representation capable of modeling the complexity of the particular data. This representation must be more reduced than representing all the observations of the phenomenon, but still describe it accurately enough. An additional problem is that information is hardly certain, complete and precise; more on the contrary, it is usually incomplete, imprecise, vague, fragmentary, not fully reliable, contradictory, or imperfect in some other ways.

Historically, two ways of addressing imperfection have been employed for representing information in a computer (Motro, 1996):

- The first solution consists in restricting the model to only that part of the available information of the real world that is accurate and reliable. Such a constrained approach avoids further complications of representation, but lacks the capacity of capturing the whole rich notion of information in human cognition and is generally very limited.
- The second solution implies developing models capable of representing imperfect information. As this approach allows a greater number of applications, it is the one that developers usually implement in their systems. However, those models cannot successfully cope with the whole range of imperfections that generally appear in real life, and in many occasions data are simplified to a point that makes them easily treatable with current computational tools, but losing part of their meaning.

Due to this lack of general systems capable of dealing with any kind of imperfect data, developers have been forced to handle this information in an ad-hoc manner; that is, by devising specific algorithms and systems for each new application, domain and representation. Therefore, in order to model the real world as accurately as possible, several approaches for dealing with imperfect information have been introduced and studied.

Managing Imperfect data

When Weaver (1948) suggested that several scientific problems were not solvable by a simple formula or probability theory, he was implying a more fundamental statement: 'almost all real-life problems cannot be solved by conventional (precise) mathematics'.

Not in vain, we experience that information in most domains is usually incomplete, imprecise, vague, fragmentary, not fully reliable, contradictory, or imperfect in some other way. Imperfect information might result from using unreliable information sources, it can be the unavoidable result of information gathering methods that require estimation or judgment, or be produced by a restricted representation model. According to Motro (1996), these various information deficiencies may result in different types of imperfection such as inconsistency, imprecision, vagueness, uncertainty, and ambiguity.

We say that we have *inconsistency* when one aspect of the real world is irreconcilably represented more than once in a data set. For example, having both '27' and '28' as values for John's age. Information inconsistency is a kind of semantic conflict that usually arises when integrating information.

Imprecision and *vagueness* are both related with the impossibility to give a concrete value to an element. The correct value is within a range of values, but there is no way of knowing which one to choose. For example, 'between 100 and 120 kilograms' and 'very heavy' for John's weight are imprecise and vague values respectively. Vague information is usually represented by linguistic terms.

Uncertainty indicates the degree of truth of a value. It expresses how sure one can be about a statement. 'It is almost sure that John is his brother' is an example of information uncertainty.

When an element of the model can have several possible interpretations, we say it is *ambiguous*. In general, if values are not accompanied by their units, it is impossible to say if a figure is high or low. A length of 1000 is meaningless unless it is stated if those are millimeters or kilometers.

Generally, several different kinds of imperfect information can coexist with respect to the same piece of information. In many real-world problems we have (or could have) statements like the following: 'it is α-certain that X is A', being X a variable, α a certainty degree, and A an imprecise value. In a statement like 'it is almost sure that John is a nice person' two sources of imperfection are present: uncertainty (for 'almost sure') and imprecision (for 'nice person').

Those two are the most common kinds of imperfection found in data. That is especially true when dealing with problems related with biological systems. Several data models have been proposed to handle uncertainty and imprecision, and most of them are based on the same paradigms. Imprecision is generally modeled with fuzzy sets Zadeh (1965), and uncertainty with fuzzy measures (evidence theory, possibility theory and probability theory).

The approach we will describe later lies within this general framework —dealing with imprecision by means of fuzzy sets, and with uncertainty by employing probability. This representation is general enough to allow its application to several problems and domains, and can be easily understood and implemented.

2.5. Representing Behaviors

Frequent Correlated Trends is our approach to tackle the problem of representing imperfect behaviors —concretely uncertain and imprecise— that came as a series of observations. This proposal is intended as a general framework that can be applied to several domains with minimum adaptation.

The present dissertation presents both a representation method and a learning algorithm suitable to be applied to data series:

- The **model** aims to summarize a particular behavior, simplifying its original representation but saving the specific attributes that differentiates that behavior from any other. Underlying local trends in the data are represented in an easy and effective way, without a complicated formalism.
- The **learning algorithm** identifies given behaviors through capturing their general footprint by means of discovering repetitive patterns in one dimension and their interdependences with patterns in other dimensions.

The model

When behaviors are defined by observations along different dimensions, *Frequent Correlated Trends* model the relationships between structural patterns in a reference dimension, and patterns in other dimensions.

As the exact values of patterns are not fully reliable because of imperfection, we employ a reduced alphabet in order to abstract those patterns and appraise their fundamental shape.

Finally, the relationships between those patterns in different dimensions are represented as sets of probability distributions.

Three characteristics of our model are of special relevance:

- It is general enough to be applied to any field in which series of observations can be found, thus not requiring complicated adaptation to particular domains or problems.
- The representation of the behavior is finite and constant in size for a given problem, regardless of the number of observations. In other words, the size of the representation does not depend on the amount of data available.
- An incremental representation is allowed and can be calculated on-line very easily. When a new value is observed, this information can be included in the representation, which is immediately updated. There is no need of recalculating the whole representation when new observations are available.

These features enable the model to automatically offer a representation of the behavior until any given observation, and allow it to deal with data series of infinite length. Because of that, *Frequent Correlated Trends* is specially aimed to represent those behaviors with a large number of observations.

The learning algorithm

In order to transform real data to the *Frequent Correlated Trends* representation, a learning algorithm is necessary. As previously said, we concentrate on behaviors that are described as multivariate data series. Therefore, the learning algorithm we propose is designed to expect that kind of input.

The first step of the algorithm consists in transforming the series of observations into series of qualitative abstract values (no more than a few) by means of a parser. This step reduces the number of different values we originally had and introduces imprecision in the data. However, that imperfection was already in the data as measures were not fully reliable. At this step, we are only making explicit that imperfection.

Once the observations have been transformed to the new reduced domain, we segment the series in small patterns. The aim is to have a reduced alphabet of possible patterns that repeat very often.

After that, we create probability distributions of patterns occurring at the same time on different dimensions. Those probability distributions provides statistical information of how patterns in different dimensions relates.

In many occasions, the final goal of representing behaviors is their posterior identification and classification. If that is the case, behaviors —represented as a collection of frequent correlated trends— are used as the patterns to compare with when a new instance is presented to the system. This comparison is done by means of a distance function that should be appropriately defined.

2.6. The application to representing music performances

Apart from describing *Frequent Correlated Trends* and its learning algorithm, we have applied that model to one particular problem: Music Performance.

In particular, we aimed to extract relevant knowledge of how a performer plays and shapes the music. Since commercial recordings are so heterogeneous, it is really difficult to exactly translate the audio to an accurate score representation; and therefore, it is impossible to map specific patterns to performers. At this point, we have already reviewed the state of the art on Computational Music Performance.

In this part of the research, we show a suitable representation method and a learning algorithm for how music is performed by different performers in terms of duration and energy. We apply such knowledge to the task of identifying the performer of new recordings.

The advantages and disadvantages that *Frequent Correlated Trends* present with respect to other methods in the literature are discussed in:

Molina-Solana, M., Arcos, J.L. & Gómez, E. (2010), "*Identifying Violin Performers by their Expressive Trends*", Intelligent Data Analysis. Vol. 14(5), pp. 555-571, DOI: 10.3233/IDA-2010-0439

The model

In this context, *Frequent Correlated Trends* propose a more abstract representation that the real notes, but still close to the melody (i.e. instead of focusing on the absolute notes, we focus on the melodic surface).

Frequent Correlated Trends characterizes, for a specific audio descriptor, the relationships a given music performer is establishing among groups of neighbor musical events. For instance, the model can qualitatively describe how changes in energy relates to a given set of consecutive ascending notes.

We represent a performance as a set of discrete probability distributions for various audio descriptors (dimensions) because the combination of trends from different audio descriptors improve the characterization.

Thus, *Frequent Correlated Trends* in this context capture statistical information about several aspects of how a certain performer tends to play.

The learning algorithm

To transform the original audio data to the *Frequent Correlated Trends* model, the first step consists in extracting audio features from the recordings. Once the notes have been estimated, along with values in several dimensions (included their duration and energy), those data series are abstracted and transformed to qualitative values.

We then segment the series in groups of three notes. In the current approach, since we segment the melodies in groups of three notes and we use two qualitative values, eight (2^3) different patterns are possible.

Finally, probability distributions are constructed for each pattern in one dimension by calculating the percentage of co-occurrence of each of those patterns with patterns in other dimensions.

Results

We worked with Sonatas and Partitas for solo violin from J.S. Bach. We tested our system by performing experiments with commercial recordings from 23 different violinists and different movements of the pieces. Each experiment consisted in learning correlated trends from one movement and then testing them with another movement.

A nearest neighbor classifier was used to generate a ranked list of possible performers for a new input recording. When a new recording is presented to the system, its *frequent correlated trends* representation is calculated and compared (by means of a distance function) to those previously learned.

In experiments using movements from the same piece , the correct performer was majority identified in the first half of the list, while in movements from different pieces, the most difficult scenario, the 90% of identification accuracy was overcame at position 15. We can also observe that a 50% of success is achieved using the five first candidates in any case (doubling the 22% of a random classifier).

These results show that our proposal is capable of learning performance patterns that are useful for distinguishing performers. The results are promising, especially comparing with a random classification where the success rate is clearly outperformed.

3. Concluding remarks

THIS chapter summarizes the contributions of this thesis to the field of Behavior Modeling and Representation, analyzing the results in accordance with the initial objectives. We also highlight the contribution that this thesis has in the field of Computational Music Performance.

3.1. Summary

Computational music performance

Studies in music performance have a particular value in our time, as it is a complex and not yet well-understood task; not in vain, the art of performing music is the result of several years of training. At the same time, contemporary information technology offers the possibility of automatic playing music specially composed for computers or stored in large databases. In such cases, the music is often played as nominally written in the score, thus implicitly ignoring the value of a living performance and its underlying art and diversity.

We have shown a wide range of research on music performance, from studies aimed at understanding expressive performance to attempts at modeling aspects of performance in a formal, quantitative and predictive way. Research on this field aims to provide expressive tools that traditionally have been hiding in musicians' skill and musical intuition. When explicitly formulated, these tools will give the user the possibility to play music files in a computer with different expressive coloring.

Throughout a review of different works, we have identified the most crucial problems that researchers must face when dealing with computational music performance. The main one is data representation, as data gathering is a complex and often unreliable task. Even more, the rules governing a performance are not exact neither well-delimited. Therefore, most efforts (ours included) are done in the direction of understanding and accurately representing how to play.

Even though we are skeptical about a machine completely replacing a human performer, we are sure that this technology will be available in a not very far future for certain tasks. Machines that could shape the music to imitate certain performers or styles will not be science-fiction products anymore. We believe that it is only a matter of time that they become commonplace. In fact, we have also shown that there are currently some attempts in this direction, like the Rencon contest.

We are strongly convinced that it is time for Computer Science to work in the music domain. Results from such research will make a great impact in both the Arts and the Sciences. Not in vain, Music is more than an interesting and, somehow, odd domain; it is part of our human essence.

Framework for music composition and performance

Focusing on the music domain, we proposed a framework for automatic music composition and performance using computers. As this is a quite challenging problem involving many tasks, a bottom-up approximation is required for solving all of them in a modular way, so that a huge problem can be broken into smaller tasks.

In order to achieve this general goal, we proposed a two-level multiagent architecture to address this issue. Our proposal also permits that users were provided by an easy-to-use interface that hides all the complexity of music composition. Even more, inputs for this interface are emotional inputs from the users, so that we are able to address the problem of music expressiveness.

We described how the system was developed and how the knowledge was represented. We also showed several agents that are already implemented, as well as their aims and roles.

Even though, *InMaMusys* is still in development, some experiments have been carried out in order to validate the approach. Results are promising enough to encourage us to continue working with this framework. We believe that Agent Theory is an adequate formalism to model the complexity of the processes of composing and performing music, and preliminary results support that claim.

Mining music motives

We presented the application of frequent pattern mining to the discovery of musical motifs in a piece of music. *MusicXML* files, which can be easily collected, are transformed into sequences of notes, defined at their lower level. Our algorithm, *SSMiner*, is able to efficiently identify the frequent subsequences within a sequence.

It is well-known that, in music, repeating patterns does not need to be exact. Our algorithm is able to identify transposed patterns, including exact matchings (i.e., null transpositions). Our experiments suggest that our approach performs well in a set of randomly-selected songs.

Frequent Correlated Trends for representing behaviors

Many phenomena in the real world can be understood as behaviors, because they follow some rules and behave in characteristic ways. The computational representation and classification of them is a task of great interest for researchers in behavioral sciences as they are in continuous need of new tools and methods to accurately represent and understand behaviors of growing complexity.

Data series are one of the representations with increasing importance lately. They are defined as an ordered sequence of data at given intervals of an indexing variable (e.g. time). Because of that, behaviors in many domains can be defined as imprecise and uncertain multivariate data series.

Although the problem of representing imperfect data has been addressed many times in the past, the lack of a general and universal solution obligates to build ad-hoc solutions for different problems. For that reason, there is still a need to find new solutions and models to represent such information.

Our proposal, which assumes that some kind of commonality exists among instances of the same behavior in a given phenomenon, represents those imperfect data series as a set of probability distributions. To do so, it first transforms the imperfect observations into qualitative values. Then, it selects a dimension of the behavior and uses it to look for correlations with the rest of dimensions. These correlations are expressed as discrete probability distributions.

The main advantage of our method is that it employs a finite and constant representation in size for behaviors, regardless of their length. It allows for an incremental representation of the observations until a particular point. Its best performance is achieved when a large number of observations is available.

Frequent Correlated Trends in music performance

The feasibility of the model was shown in the domain of music performances. We focused on the task of identifying violinists from their playing style by means of using frequent correlated trends to capture expressive tendencies. The way of play can be understood as a behavior as we have previously defined it.

The learning process departs from state-of-the-art audio feature extraction tools, and an automatic segmentation of the melodies using IR patterns (Narmour, 1990). Performers are characterized by a set of frequency distributions, capturing their personal style with respect to a collection of melodic patterns. We have shown that, without a great analysis accuracy, our proposal is quite robust for representing the particular nuances of different performers.

Experiments concentrated on identifying violinists by using note durations and energies as descriptors. We tested the system with 23 different professional performers and different recordings. Obtained results showed that the *Frequent Correlated Trend* model is capable of learning performance patterns that are useful for distinguishing performers. The results clearly outperform a random classifier and, probably, it would be quite hard for human listeners to achieve such recognition rates.

Contributions

The initial objectives has been completely fulfilled in the following ways:

Understanding music performances. We made a review of different approaches for understanding and representing music performances, mainly from a computational point of view. The result can be found in page 45.

Application of DataMining to the music domain. As part of our study on computational tools applied to the music domain, we developed an algorithm for discovering frequent patterns in compositions, focusing on identifying the chorus of different songs. We fulfilled this objective in page 61.

Development of a framework. The framework for integrating all the learning tools in a music context, is described in the paper at page 53.

Representation of imprecise and uncertain behaviors. This objective is pursued in page 79. We described what behaviors are and we proposed an abstract representation model for imprecise and uncertain behaviors.

Application to represent music performances. Finally, we tested that model in a particular field: the representation of music performances. This application, along with its results is presented in the paper at page 107.

3.2. Further research

Departing from the work done so far, further research can be performed in several directions. We described them organized by the topics covered in this dissertation.

Multiagent framework for music composition and performance

Our current *InMaMuSys* prototype is quite rigid in the way it deals with user inputs. It does not take advantage of fuzzy techniques to address emotions and musical terms. So that, including fuzzy logic in the inference system and the interface is one of the changes to be done. Despite the major modifications, this change will allow *InMaMuSys* to better deal with the complexity of the domain, providing more flexible responses to user inputs.

Secondly, we are interested in developing a greater number of agents in order to get a bigger collection and cover a wider range of compositional styles. Our aim is to obtain as much diversity as possible not only to compare different algorithms and compositional mechanisms, but to implement and test new ideas. These ideas could be related with new theories of musical styles or cognitive processes. Even more, as the set of composers growths, we will be able to reproduce a wider range of human emotions.

Another interesting enhancement is that of developing a module to automatically obtain composition rules. In the current prototype, this knowledge is given by humans, and coded into the agents. The system would dramatically improve its capacities and autonomy if it could analyze a music sheet, extract some rules, and finally compose according to them. Doing so, the system would be quite complete, in the sense that it would include both composition and analysis tasks.

Mining music motives

Regarding this issue, we intend to employ interval strings to represent melodies rather than the absolute pitches we have used in the experiments. As melodic structures are mainly intervalic, this change will greatly enhance the performance of the algorithm.

We will also consider more abstract representations of melodies, based on contours, such as the one proposed by Narmour (1990). Again, we would like to take advantage of a particular characteristic of music to improve the search: repetitions are hardly exactly equal. Employing an appropriate abstract representation will result in small differences encoded as equal, and thus improving the performance of the searching process.

Finally, we plan to study the parallelization of our algorithm implementation in order to improve its execution time, which is already asymptotically optimal.

Frequent Correlated Trends

Regarding our model for representing behaviors, the first line of further work consists in applying the *Frequent Correlated Trend* model to different domains in order to test their feasibility. Some of them has been described in this dissertation as illustrative examples, but many others can be candidates.

Besides that, we plan to employ other fuzzy measures apart from probability theory (specifically possibility theory). Not in vain, both probability and possibility theories are suitable for modeling uncertainty, but each one excels in different types. As our framework aims to be a general one, addressing uncertainty in all their varieties is hence a requisite. Obviously, for each problem it has to be decided which theory is more appropriate to represent the semantic of the observations.

A different issue would be the study of a distance measure appropriate for each domain. We have initially defined a simple one that performs well, but many others can be proposed to better account for the particular semantics of different applications.

Frequent Correlated Trends in music performance

Regarding the application of frequent correlated trends to the task of representing music performances, we would like to analyze the validity of the representation when using several qualitative values and not only the current two (greater and lesser).

We foresee a stronger use of fuzzy values to represent measures and those values. This formalism could also be employed to represent the underlying histograms. This improvement will allow a better assessment in the similarity measure.

Combining information from different music features has been demonstrated to improve results. We are currently working on increasing the number of descriptors. Since the predictability of a given descriptor varies depending on the performers, we are also interested in discovering relations among the descriptors that could allow us to reduce the dimensionality of the representation.

We also acknowledge that the use of hierarchical classifiers and ensemble methods is a possible way to improve the identification accuracy, as they will be able to selectively choose the best alternative to distinguish between one class and the rest.

Computational music performance

Finally, we would like to reflect on the future goals that the computational music performance field is to achieve. They were adequately summarized by the S2S2 Consortium in its document (2007) from the 6th Framework Programme in the Future and Emergent Technologies. Those goals are challenging enough to attract researching efforts, being the field a very active one, with plenty of room for new research in the area.

Since the literal synthesis of notes from a score is bland and unappealing, there is an opportunity for learning systems that can automatically produce compelling expressive variations. The problem of synthesizing expressive performance is both exciting and challenging. Musical performance is one of the many activities that trained people do very well without knowing exactly how they do it. This is, precisely, one of the main problems to be faced because there is no model that accurately tells us how to perform.

When referring to artistic domains, it is hardly possible to find a 'correct' model whose predictions always correspond with what humans do and what they think is acceptable. We cannot forget that evaluation in these domains is often subjective and heavily-dependent on who is speaking.

Many aspects are involved within expressive performance and it is almost impossible to use them all. Moreover, there are some parameters and dimensions which are commonly considered as non-relevant but that, in fact, might be. Only a portion of the whole problem is tackled by current techniques. One future challenge is to address the problem by using as much dimensions as possible, as it could be possible that some important patterns are hidden and we haven't still discovered them.

Moreover, to obtain very precise data about all those parameters is a challenging problem that cannot still be done in a automatic way. Annotating all this information is an extremely time-consuming task and requires a lot of effort from several humans. Early systematic investigations in the field have dealt with this problem either by reducing the length of the music (to just some seconds) or by controlling the size of the collections.

Recent approaches try to avoid this task by the use of some statistical learning techniques and by focusing in a more abstract representation of the real notes and their values. Statistical musicology has not historically received much attention, but it is increasing its popularity as the amount of available data grows, even though collecting large amount of quantitative data is still a hard task.

Despite some successes in computational performance modeling, current models are extremely limited and simplistic regarding the complex phenomenon of musical expression. It remains an intellectual and scientific challenge to probe the limits of formal modeling and rational characterization. Clearly, it is strictly impossible to arrive at complete predictive models of such complex human phenomena. Nevertheless, work towards this goal can advance our understanding and appreciation of the complexity of artistic behaviors. Understanding music performance will require a combination of approaches and disciplines, such as musicology, AI, machine learning, psychology and cognitive science.

For cognitive neuroscience, discovering the mechanisms which govern the understanding of music performance is a first-class problem. Different brain areas are involved in the recognition of different performance features. Knowledge of these can be an important aid to formal modeling and rational characterization of higher order processing, such as the perceptual differentiation between human-like and mechanical performances. Since music making and appreciation is found in all cultures, the results could be extended to the formalization of more general cognitive principles.

Finally but not least, it is the problem of the individuality of each work. Even though there is a huge amount of available data, every song is different from the rest. Hence, it would not be adequate just to apply the way of playing Beethoven's Ninth Symphony to Brahms' Symphonies. A deep study of the work is needed in order to understand the author, the context and the music. One should always keep in mind that artistic performance is far from being predictable.

Bibliography

Rakesh Agrawal and Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In *20th International Conference on Very Large Data Bases (VLDB'94)*, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc. ISBN 1-55860-153-8. 18

Fernando Alonso, África López-Illescas, Loïc Martínez, César Montes, and Juan P. Valente. Knowledge discovery using medical data mining. In Alfredo Colosimo, Paolo Sirabella, and Alessandro Giuliani, editors, *Medical Data Analysis*, volume 2526 of *Lecture Notes in Computer Science*, pages 1–12. Springer Berlin / Heidelberg, 2002. 21

Josep Lluis Arcos, Ramón López de Mántaras, and Xavier Serra. SaxEx: a case-based reasoning system for generating expressive musical performances. *Journal of New Music Research*, 27(3):194–210, 1998. doi: 10.1080/09298219808570746. 15

Shlomo Argamon, Kevin Burns, and Shlomo Dubnov, editors. *The Structure of Style*. Springer-Verlag, 1st edition, 2010. doi: 10.1007/978-3-642-12337-5. 5

Jean-Julien Aucouturier and Mark Sandler. Finding repeating patterns in acoustic musical signals: Applications for audio thumbnailing. In *Audio Engineering 22nd International Conference on Virtual, Synthetic and Entertainment Audio (AES22)*, pages 412–421, 2002. 18

Tengfei Bao, Huanhuan Cao, Enhong Chen, Jilei Tian, and Hui Xiong. An unsupervised approach to modeling personalized contexts of mobile users. *Knowledge and Information Systems*, 31(2):345–370, 2012. ISSN 0219-1377. doi: 10.1007/s10115-011-0417-1. 21

Roberto Bresin and Anders Friberg. Emotional coloring of computer-controlled music performances. *Computer Music Journal*, 24(4):44–63, December 2000. doi: 10.1162/014892600559515. 12

Peter J. Brockwell and Richard A. Davis. *Time Series: Theory and Methods*. Springer, 1991. 21

Emilio Cambouropoulos, Tim Crawford, and Costas S. Iliopoulos. Pattern processing in melodic sequences: Challenges, caveats and prospects. *Computers and the Humanities*, 35(1):9–21, 2001. 18

Longbing Cao. In-depth behavior understanding and use: The behavior informatics approach. *Information Sciences*, 180(17):3067–3085, 2010. ISSN 0020-0255. doi: 10.1016/j.ins.2010.03.025. 20

Miguel Delgado, Waldo Fajardo, and Miguel Molina-Solana. INMAMUSYS: Intelligent multiagent music system. *Expert Systems with Applications*, 36(3):4574–4580, 2009a. doi: 10.1016/j.eswa.2008.05.028.

Miguel Delgado, María Ros, and M. Amparo Vila. Correct behavior identification system in a tagged world. *Expert Systems with Applications*, 36:9899–9906, 2009b. doi: 10.1016/j.eswa.2009.01.077. vii, 5, 21

Miguel Delgado, Waldo Fajardo, and Miguel Molina-Solana. A state of the art on computational music performance. *Expert Systems with Applications*, 38(1):155–160, 2011. doi: 10.1016/j.eswa.2010.06.033.

Matthew J. Dovey. Analysis of Rachmaninoff's piano performances using inductive logic programming. In *Proceedings of the Eighth European Conference on Machine Learning (ECML95)*, pages 279–282, London, UK, 1995. Springer-Verlag. ISBN 3-540-59286-5. 13

Anders Friberg. Generative rules for music performance: A formal description of a rule system. *Computer Music Journal*, 15(2):56–71, 1991. 15, 16

Anders Friberg, Vittorio Colombo, Lars Frydén, and Johan Sundberg. Generating musical performances with director musices. *Computer Music Journal*, 24(3):23–29, 2000. 15

Alf Gabrielsson. *Music and the Mind Machine*, chapter Expressive intention and performance, pages 35–47. Springer-Verlag, Berlin, Germany, 1995. 12

Mitsuyo Hashida, Keiji Hirata, and Haruhiro Katayose. Rencon Workshop 2011 (SMC-Rencon): Performance Rendering Contest for Computer Systems. In *Eighth Sound and Music Computing Conference (SMC 2011)*, Padova, Italy, July 2011. 15

Ju-Lee Hong. Investigating expressive timing and dynamics in recorded cello. *Psychology of Music*, 31(3):340–352, 2003. 13

Osamu Ishikawa, Yushi Aono, Haruhiro Katayose, and Seiji Inokuchi. Extraction of musical performance rules using a modified algorithm of multiple regression analysis. In *Proceedings of the 2000 International Computer Music Conference*, pages 348–351, San Francisco, USA, 2000. 15

Aida Jiménez, Fernando Berzal, and Juan Carlos Cubero. POTMiner: Mining Ordered, Unordered, and Partially-Ordered Trees. *Knowledge and Information Systems*, 23(2): 199–224, 2010. doi: 10.1007/s10115-009-0213-3. 18

Aida Jiménez, Miguel Molina-Solana, Fernando Berzal, and Waldo Fajardo. Mining transposed motifs in music. *Journal of Intelligent Information Systems*, 36(1):99–115, 2011. doi: 10.1007/s10844-010-0122-7.

Patrik N. Juslin, Anders Friberg, and Roberto Bresin. Toward a Computational Model of Expression in Performance: the GERM Model. *Musicae Scientiae*, special issue: 63–122, 2002. 13

Elmar Kriegler and Hermann Held. Utilizing belief functions for the estimation of future climate change. *International Journal of Approximate Reasoning*, 39:185–209, 2005. doi: 10.1016/j.ijar.2004.10.005. 21

Fred Lerdahl and Ray Jackendoff. *A generative theory of tonal music*. Cambridge, Mass: MIT Press, 1983. 16

Mark Levy and Mark Sandler. Structural segmentation of musical audio by constrained clustering. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2): 318–326, 2008. doi: 10.1109/TASL.2007.910781. 18

Ramón López de Mántaras and Josep Lluis Arcos. AI and Music: from Composition to Expressive Performance. *AI Magazine*, 23(3):43–57, 2002. 13, 16

M. Marques, V. Oliveira, and A.C. Rosa. Music composition using genetic evolutionary. In *2000 Congress on Evolutionary Computation*, pages 714–719, 2000. 16

David Meredith, Kjell Lemström, and Geraint A. Wiggins. Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research*, 31(4):321–345, 2002. 18

Marvin Minsky. *Machine models of music*, chapter Music, Mind, and Meaning, pages 327–354. MIT Press, Cambridge, MA, USA, 1992. ISBN 0-262-19319-1. 15

Eduardo R. Miranda. At the crossroads of evolutionary computation and music: Self-programming synthesizers, swarm orchestras and the origins of melody. *Evolutionary Computation*, 12(2):137–158, 2004. 16

Miguel Molina-Solana, Josep Lluis Arcos, and Emilia Gómez. Identifying violin performers by their expressive trends. *Intelligent Data Analysis*, 14(5):555–571, 2010. vii, 5, 21

Amihai Motro. Sources of Uncertainty, Imprecision, and Inconsistency in Information Systems. In Amihai Motro and Philippe Smets, editors, *Uncertainty Management in Information Systems: From Needs to Solutions*, pages 9–34. Kluwer Academic Publishers, 1996. 22, 23

Eugene Narmour. *The Analysis and Cognition of Basic Melodic Structures: The Implication Realization Model*. Chicago, IL: Univ. Chicago Press, 1990. 32, 34

Jouni Paulus and Anssi Klapuri. Music structure analysis using a probabilistic fitness measure and a greedy search algorithm. *IEEE Trans. on Audio, Speech, and Language Processing*, 17(6):1159–1170, 2009. 18

Marcus T. Pearce, David Meredith, and Geraint A. Wiggins. Motivations and methodologies for automation of the compositions process. *Musicae Scientiae*, 6(2):119–147, 2002. 17

Anna Pienimäki. Indexing music databases using automatic extraction of frequent phrases. In *3rd International Conference on Music Information Retrieval (ISMIR 2002)*, pages 25–30, 2002. 18

Giovanni De Poli. Methodologies for expressiveness modelling of and for music performance. *Journal of New Music Research*, 33(3):189–202, 2004. 13

Melvin G. Rigg. The mood effects of music: a comparison of data from former investigators. *Journal of Psychology*, 58:427–438, 1964. 12

Pierre-Yves Rolland. Discovering patterns in musical sequences. *Journal of New Music Research*, 28(4):334–350, 1999. 18

Craig Sapp. Comparative analysis of multiple musical performances. In *8th International Conference on Music Information Retrieval (ISMIR 2007)*, pages 497–500, Vienna, Austria, 2007. 14

Craig Saunders, David Hardoon, John Shawe-Taylor, and Gerhard Widmer. Using string kernels to identify famous performers from their playing style. *Intelligent Data Analysis*, 12(4):425–440, 2008. 14

Efstathios Stamatatos and Gerhard Widmer. Automatic identification of music performers with learning ensembles. *Artificial Intelligence*, 165(1):37–56, 2005. 13

Johan Sundberg. Emotive transforms. *Phonetica*, 57:95–112, 2000. 12

The S2S2 Consortium. *A Roadmap for Sound and Music Computing*. The S2S2 Consortium, 2007. ISBN 9789081189613. URL http://smcnetwork.org/roadmap. 35

Neil P. Todd. The dynamics of dynamics: A model of musical expression. *Journal of the Acoustical Society of America*, 91(6):3540–3550, 1992. 13, 15

Warren Weaver. Science and Complexity. *American Scientist*, 36:536–544, 1948. 23

Gerhard Widmer. Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries. *Artificial Intelligence*, 146(2):129–148, 2003. 15

Gerhard Widmer and Werner Goebl. Computational models of expressive music performance: The state of the art. *Journal of New Music Research*, 33(3):203–216, 2004. 12, 13

Gerhard Widmer, Simon Dixon, Werner Goebl, Elias Pampalk, and Asmir Tobudic. In search of the Horowitz factor. *AI Magazine*, 24(3):111–130, 2003. 13

Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965. 24

Mohammed J. Zaki. Efficiently mining frequent trees in a forest: Algorithms and applications. *IEEE Trans. on Knowledge and Data Engineering*, 17(8):1021–1035, 2005. doi: 10.1109/TKDE.2005.125. 18

Patrick Zanon and Giovanni De Poli. Estimation of parameters in rule systems for expressive rendering in musical performance. *Computer Music Journal*, 27(1):29–46, 2003. 15

Qijun Zhang and Eduardo R. Miranda. Towards and evolution model of expressive music performance. In *Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06)*, pages 1189–1194, 2006. 16

Tong Zhang and Ramin Samadani. Automatic generation of music thumbnails. In *Proceedings of the 2007 IEEE International Conference on Multimedia and Expo*, pages 228–231, 2007. doi: 10.1109/ICME.2007.4284628. 20

Yan-Qing Zhang and Xuhui Wan. Statistical fuzzy interval neural networks for currency exchange rate time series prediction. *Applied Soft Computing*, 7(4):1149–1156, 2007. doi: 10.1016/j.asoc.2006.01.002. 21

Part II.

Publications

A state of the art on computational music performance

Delgado, M., Fajardo, W. & Molina-Solana, M. (2011), "*A state of the art on computational music performance*", Expert Systems with Applications. Vol. 38(1), pp. 155-160, DOI: 10.1016/j.eswa.2010.06.033

- Status: **Published**
- Impact Factor (JCR 2010): 1.926
- Subject category:
    - COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE (34/108 Q2)
    - ENGINEERING, ELECTRICAL & ELECTRONIC (50/247 Q1)
    - OPERATIONS RESEARCH & MANAGEMENT SCIENCE (15/75 Q1)

Expert Systems with Applications 38 (2011) 155–160

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A state of the art on computational music performance

Miguel Delgado, Waldo Fajardo, Miguel Molina-Solana *

Department of Computer Science and Artificial Intelligence, Universidad de Granada, Daniel Saucedo Aranda s/n, 18071 Granada, Spain

ARTICLE INFO

Keywords:
Computational music
Expressive performance
Machine learning

ABSTRACT

Musical expressivity can be defined as the deviation from a musical standard when a score is performed by a musician. This deviation is made in terms of intrinsic note attributes like pitch, timbre, timing and dynamics. The advances in computational power capabilities and digital sound synthesis have allowed real-time control of synthesized sounds. Expressive control becomes then an area of great interest in the sound and music computing field. Musical expressivity can be approached from different perspectives. One approach is the musicological analysis of music and the study of the different stylistic schools. This approach provides a valuable understanding about musical expressivity. Another perspective is the computational modelling of music performance by means of automatic analysis of recordings. It is known that music performance is a complex activity that involves complementary aspects from other disciplines such as psychology and acoustics. It requires creativity and eventually, some manual abilities, being a hard task even for humans. Therefore, using machines appears as a very interesting and fascinating issue. In this paper, we present an overall view of the works many researchers have done so far in the field of expressive music performance, with special attention to the computational approach.

1. Introduction

Imagine the scene. You switch on your hi-fi system, select *Clair de Lune* by Claude Debussy and Sergei Rachmaninoff as performer. After that, you hit the 'Play' button, sit on your favourite armchair and enjoy the music. The situation sounds perfectly normal ... but for a small detail: Rachmaninoff never recorded Debussy's *Clair de Lune*!

To listen to a performance we need a performer. So far, this role has always been assumed by humans but, why can't the hi-fi system (more generally, a computer) be the performer and play the music as it was Rachmaninoff himself? All it needs is enough knowledge of how to play.

As Widmer, Dixon, Goebl, Pampalk, and Tobudic (2003) stated, when skilled musicians play a piece of music, they do not do it mechanically, with constant tempo or loudness, exactly as written in the printed music score. Rather, they speed up at some places, slow down at others and stress certain notes. The most important parameters available to a performer are timing (tempo variations) and dynamics (loudness variations). The way these parameters 'should be' varied during the performance is not precisely specified in the printed score. So that, it is performer's duty to use them properly.

It is a fact that student musicians spend more time practicing than almost any other activity. Weekly music lessons, endless scales, nightly rehearsals and recitals for friends and family are commonplace in their lives. Hours of practicing will help them learn to interpret a piece of music as the composer envisioned it, as well as to develop their own signature sound – one that is unique to each of them. In other words, what makes a piece of music come alive is also what distinguishes great artists from each other.

Other questions arise at this point: how should those expressive resources be employed? What is that which makes Rachmaninoff an outstanding pianist? And those simple questions, which many people have asked for many years, do not have still a clear answer from musicologists. Even when those questions will eventually find an acceptable answer, another will be posed: can a computer take advantage of that knowledge, being able to substitute a famous performer? As we will see in this paper, many attempts have been made and several computational models have been proposed during the last century to do so.

This work is organized as follows: first of all, Section 2 describes what is musical performance and its parameters, and how they can be used to distinguish between performers; Section 3 presents some works where computers were used for extracting information about those parameters from the music itself, for representing that knowledge in a computer, and for applying it to generate new

* Corresponding author.
E-mail addresses: mdelgado@ugr.es (M. Delgado), aragorn@ugr.es (W. Fajardo), miguelmolina@ugr.es (M. Molina-Solana).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.06.033

performances; Section 4 introduces some difficulties to be faced in the future when using computers in this domain; and Section 5 summarizes the work.

2. Music performance

Most people would judge the literal execution of a musical score to be significantly less interesting than a performance of that piece by even a moderately skilled musician. Why is that so? Because what we hear is not a literal rendition of the score. Of course, the principal vehicle for the communication of musical compositions is the music score in which the composer codifies his intentions. However, the information written in the score does not represent an exhaustive description of the composer's intentions. It carries information such as the rhythmical and melodic structure of a certain piece, but there is not yet a notation able to describe precisely the timing and timbre characteristics of the sound.

When speaking, we use several voice resources such as changing velocity, tone or loudness. All this effects are not explicitly in the text we are reading. In fact, when several people read a text, resulting sounds are not the same, even though words in the sheet remain unchanged. So does in music. In the same way that in a written poem there is no explicit notation for how to pronounce, in musical scores there is also such a lack of information. This comparison is actually quite appropriate because former research on music performance has revealed interesting analogies in the communication of emotions in singing and speech (Bresin & Friberg, 2000; Sundberg, 2000).

Performing is a crucial activity in music. In many kinds of music the performer acts as a kind of mediator: a mediator between composer and listener, between written score and musical sound. It is the performer who renders each note in the score in terms of intensity, duration and timbre by movements of fingers, arms or mouth. This results in different performances of the same piece reflecting each performer's culture, mood, skill and intention. These variances also contribute to determining the performing styles of different musicians. So that, the music we hear has two main sources: the score and the performance, and they both need from the other.

Briefly, Widmer and Goebl (2004) define *expressive music performance* as ''the deliberate shaping of the music by the performer, in the moment of playing, by means of continuous variations of parameters such as timing, loudness or articulation''. Changes in tempo (timing) are non-linear warping of the regular grid of beats that defines time in a score. It is also possible to change only the duration of certain notes. Changes in loudness (or dynamics) are modifications of the intensity of notes with respect to the others and to the general energy of the fragment in consideration. Articulation consists in varying the gap between contiguous notes by, for instance, making the first one shorter or overlapping it with the next.

Music performance is a deep human activity which requires emotional, cognitive and artistic aptitudes. At the same time, it is also a complex task involving physical, acoustic, physiological, psychological, social and artistic aspects. Several factors determine the rendition of a musical piece. One of the most obvious is the physical condition of the performer. Not in vain, performer's mood, health and fatigue play a crucial role in the process of playing an instrument. Some studies (see those from Gabrielsson (1995) and from Rigg (1964)) have shown major variations in renditions by the same performer when he is in different moods.

Manual abilities are also an important point that is especially visible when comparing a beginner with an expert. With practice, musicians can improve their velocity and precision, reducing the amount of unintended deviations with respect to the score (commonly known as errors). Other factors that affect the rendition are the location where it takes place and the instrument being used. The acoustics of the place are important because they establish the sounds that can be made. So does the instrument, which has an evident influence on the character of the work.

Because the conventional score is quite inadequate to describe the complexity of a musical performance, and since the literal synthesis of notes from a score is flat and unappealing, there is an opportunity for learning systems that can automatically produce compelling expressive variations. Hence, methods for automatically ''bringing life'' to musical scores become useful and interesting. Research in this field ranges from studies aimed at understanding expressive performance to attempts at modelling aspects of performance in a formal, quantitative and predictive way, so that a computer might be able to perform them.

2.1. Functions of expressivity

Since the very first moment that some deviations exist in the way of playing a score, we can ask for the motives of their existence. Two main aims can be identified in a first sight.

In first place, expressivity is used as an instrument for communicating emotions. Meyer (1956) stated that meaning (be it emotional or aesthetic) arises in music when expectations raised by the music are not realized. It was Rigg's paper (Rigg, 1964) one of the pioneer works which tackled the relation between emotions and musical structure. Some interesting and typical regularities found throughout the years were described there: solemn music tend to be slow, low pitched, and without irregularities; happy music is fast, major mode and high pitched.

Gabrielsson (1995) and Lindström (2006) studied the relation between motional intentions and musical microstructure (for instance, tempo deviations, changes in intensity or articulations). Canazza, Poli, Drioli, Rodà, and Vidolin (2000) studied how physical parameters in musical recordings (tone, articulations or global tempo) were affected by the modification of performer's expressive intentions. In their experiments, the performer was asked to express, by her rendition of the musical score, sensorial concepts such as 'bright', 'light' or 'dark'. The sonological analysis of the recordings made it possible to relate certain values to given concepts (e.g., a 'light rendition' was found to be in fast tempo, with shortened note durations and soft attacks).

The significance of various performance parameters in the identification of emotional qualities of a performance has been tested in synthesis experiments. Automatic performances were obtained by setting certain expressive cues to greater or lesser values and, in formal listening tests, listeners were able to recognize and identify the intended emotions. In the computer program developed by Canazza et al. expressiveness was applied both to a 'neutral' performance played by a musician with no intended emotion, and to a computer-generated 'deadpan' performance. Juslin (1997), on the other hand, manually adjusted the values of some previously identified cues by means of ''appropriate settings on a Roland JX1 synthesizer that was MIDI-controlled'' by a Synclavier III.

For more information regarding research in musical performance, including the role expressivity plays in the communication of emotions, Gabrielsson's work (Gabrielsson, 2003) might be consulted.

In second place, expressivity clarifies the musical structure, understanding within this term the metrical structure, phrasing and harmonic structure. In the work by Sloboda (1983), one could observe that performers tend to play louder and more legato the notes at the beginning of measures. It was also reported that the more expert the pianist was, the more frequent those resources were employed and the easier to transcribe the music for the audience.

Musical structure has its influence on the expressivity of performances too. It has been discovered that the beginning and the end of phrases tend be slower than the rest. For instance, Todd (1989) proposed a model to predict the final *rubato* in musical works.

Harmonic progressions in a work also have an influence on the expressivity of its renditions. In particular, Palmer (1996) demonstrated that melodic expectation —the degree in which an expected note is finally realized— was related to the energy with which notes are played.

3. Computational music performance achievements

Advances in digital sound synthesis and in computational power have enabled real-time control of synthesized sounds. Expressive control of these becomes then a relevant area of research in the *Sound and Music Computing*[1] field. Empirical research on expressive music performance has its origin in the 1930s, with the pioneering work by Seashore (1938). After a 40-years gap, the topic experienced a real renaissance in the 1970s, and music performance research is now highly productive. A comprehensive overview of this research can be found in Gabrielsson (2003).

As said before, research in musical performance has a multidisciplinar character, with studies that veer from understanding expressive behaviour to modelling aspects of renditions in a formal quantitative and predictive way. Historically, research in expressive music performance has focused on finding general principles underlying the types of expressive 'deviations' from the musical score (e.g., in terms of timing, dynamics and phrasing) that are a sign of expressive interpretation. Works by Poli (2004) and Widmer and Goebl (2004) contains recent overviews on expressive performance modelling.

Three different research strategies can be distinguished: (1) acoustic and statistical analysis of performances by real musicians –the so-called analysis-by-measurement method; (2) making use of interviews with expert musicians to help translate their expertise into performance rules –the so-called analysis-by-synthesis method; and (3) inductive machine learning techniques applied to large databases of performances.

Studies by several research teams around the world have shown that there are significant regularities that can be uncovered in these ways, and computational models of expressive performance (of mostly classical music) have proved to be capable of producing truly musical results. These achievements are currently inspiring new research into more comprehensive computational models of music performance and also ambitious application scenarios.

One of the issues in this area is the representation of the way certain performers play by just analyzing some of their renditions (i.e., study the individual style of famous musicians). That information would enable us to identify a performer by only listening to their rendition. These studies are difficult because the same professional musician can perform the same score in very different ways (compare several commercial recordings by Sergei Rachmaninoff or Vladimir Horowitz). Recently, new methods have been developed for the recognition of music performers and their style. Among them, the most relevant are the fitting of performance parameters in rule-based performance models, and the application of machine learning methods for the identification of performing style of musicians. Recent results of specialized experiments show surprising artist recognition rates (for instance, see those from Saunders, Hardoon, Shawe-Taylor, & Widmer, 2008; or Molina-Solana, Arcos, & Gomez, 2008).

So far, music performance research has been mainly concerned with describing detailed performance variations in relation to musical structure. However, there has recently been a shift towards high-level musical descriptors for characterizing and controlling music performance, especially with respect to emotional characteristics. For example, it has been shown that it is possible to generate different emotional expressions of the same score by manipulating rule parameters in systems for automatic music performance (Bresin & Friberg, 2000).

Interactive control of musical expressivity is traditionally a conductor's task. Several attempts have been made to control the tempo and dynamics of a computer-played score with some kind of gesture input device. For example, Friberg (2006) describes a method for interactively controlling, in real-time, a system of performance rules which contains models for phrasing, micro-level timing, articulation and intonation. With such systems, high-level expressive control can be achieved. Dynamically controlled music in computer games is another important future application.

Recently, some efforts have been made in the direction of visualizing expressive aspects of music performance. Langner and Goebl (2003) have developed a method for visualizing expressive performances in a tempo-loudness space: expressive deviations leave a trace on the computer screen in the same way as a worm does when it moves, producing a sort of 'fingerprint' of the performance. This method has been recently extended by Grachten, Goebl, Flossmann, and Widmer (2009). This and other recent methods of visualization can be used for the development of new multi-modal interfaces for expressive communication, in which expressivity embedded in audio is converted into visual representation, facilitating new applications in music research, music education and Human–Computer Interaction, as well as in artistic contexts. A visual display of expressive audio may also be desirable in environments where audio display is difficult or must be avoided, or in applications for hearing-impaired people.

For many years, research in Human–Computer Interaction in general and in sound and music computing in particular was dedicated to the investigation of mainly 'rational' abstract aspects. In the last ten years, however, a great number of studies have emerged which focus on emotional processes and social interaction in situated or ecological environments. The broad concept of 'expressive gesture', including music, human movement and visual (e.g., computer animated) gesture, is the object of much contemporary research.

3.1. Data acquisition

In this interdisciplinary research field, the obtention of information on musical expressivity can be approached from different perspectives. One approach is the musicological analysis of music and the study of the different stylistic schools. This approach provides a valuable understanding about musical expressivity.

Another perspective is the computational modelling of music performance by means of automatic analysis of recordings. This sound analysis perspective can be raised by the (studio specific) recording of several performers where several expressive resources are emphasized. That information can be gathered by using augmented instruments (i.e., instruments provided with sensors of pressure or movement). Proceeding this way, the data on obtains is very precise, but it is necessary a complex setup and those special instrument are anything but cheap. Furthermore, getting the performers is a difficult task and many times even impossible (e.g. dead performers).

An alternative approach is to directly use commercial recordings for the analysis of expressivity, extracting all the relevant data from the audio signals themselves. This approach has several advantages: there are tons of recordings available (and often some performers have several ones); and the performances are 'real' and gather the decisions taken by the performers without any external

[1] http://smcnetwork.org.

influence. Nevertheless, working with commercial recordings has some important drawbacks too: some information (consider, for instance, the bow speed in a violin) cannot be easily gained from the audio; these recordings do not come from a controlled scenario and the sound analysis may become more difficult.

Computers are important in both approaches, because they allow us to store and to process all the gathered data. This information is huge in size and it is impossible to deal with it in a manual way.

3.2. Computational models for artistic music performance

The use of computational music performance models in artistic contexts (e.g., interactive performances) raises a number of issues that have so far only partially been faced. The concept of a creative activity being predictable and the notion of a direct 'quasi-causal' relation between the musical score and a performance are both problematic. The unpredictable intentionality of the artist and the expectations and reactions of listeners are neglected in current music performance models. Surprise and unpredictability are crucial aspects in an active experience such as a live performance. Models considering such aspects should take account of variables such as performance context, artistic intentions, personal experiences and listeners' expectations.

In the past, this problem has been tackled by using machine learning techniques. For instance, Juslin, Friberg, and Bresin (2002) described the main sources of expressivity in musical renditions and expressed the necessity of integrating some of this aspects in a common model they started to sketch.

Ramírez, Maestre, Pertusa, Gómez, and Serra (2007) proposed a model for identifying saxophonists from the way of playing by using very precise information about deviations in parameters such as pitch, duration and loudness. They measure those deviations both in inter and intra note level.

De Mántaras and Arcos (2002) studied the expressivity of several AI-based systems for music composition. They compared this expressivity with the one that exists in human recordings. Moreover, they introduced *SAXEX*, a system capable of generating expressive performances of jazz ballads by using examples from human performers and a case-based reasoner.

Hong, on the other hand, studied how musical expressivity is affected by tempo and dynamics variations (Hong, 2003). He employed cello recordings for the experiments. He extended previous work by Todd (1992), by applying new musical ideas from the 20th century to Todd's model.

Dovey (1995) proposed an attempt to use inductive logic in order to determine the rules that pianist Sergei Rachmaninoff may have used in their performances with an augmented piano. The aim was to extract general rules (in the form of universal predicates) about each note's duration, tempo and pressure. All that information was obtained from the way of playing the piano.

The group led by Gerhard Widmer has worked in the automatic identification of pianists. In Widmer et al. (2003), they studied how to measure several aspects of performances by applying machine learning techniques; whereas in another work (Stamatatos & Widmer, 2005), they proposed a set of simple features that could serve to represent performer's expressivity from a rendered musical work.

Moreover, in a recent paper, Saunders et al. (2008) represent musical performances as string of symbols from an alphabet. Those symbols contain information about changes in timing and energy within the song. After that, they use *Support Vector Machines* to identify the performer in new recordings.

Sapp's work is also an interesting proposal, as it represents musical renditions by means of sketches which are based on the correlation between time and energy (Sapp, 2007).

Most of the modelling attempts in performance research, try to capture common performance principles, that is, they focus on commonalities between performances and performers. However, the ultimate goal of this kind of research and of many of the works is not the automatic style replication or the creation of artificial performers, but to use computers to teach us more about the elusive artistic activity of expressive music performance. While it is satisfying to see that the computer is indeed capable of extracting information from performance measurements that seems to capture aspects of individual style, this can only be a first step. In order to get real insight, we will need learning algorithms that, unlike nearest-neighbour methods, produce interpretable models.

Although it may sound odd, there are concrete attempts at elaborating computational models of expressive performance to a level of complexity where they are able to compete with human performers. The Rendering Contest (Rencon)[2] (Hiraga, Bresin, Hirata, & Katayose, 2004) is an annual event first launched in 2002. It tries to bring together scientist from all over the world for a competition of artificially created performances. It uses an human judge to evaluate music performances automatically generated by computers. Participants are asked to generate a rendition of a musical work by using a predictive level. In a wider sense, we can somehow see this paradigm as an expressive performance *Turing test*.[3] In other words, the best systems are those than manage to generate performances which sounds indistinguishable from human ones.

As can be seen, music performance is an interesting research topic which enables the study of human's emotions, intelligence and creativity. These are precisely the issues Marvin Minsky referred to when he wrote about music as a human activity (Minsky, 1992).

3.3. Automatic music performance

The principal characteristic of an automatic performance system is that it converts a music score into an expressive musical performance typically including time, sound and timbre deviations from a deadpan realization of the score. Mostly, two strategies have been used for the design of performance systems, the analysis-by-synthesis method and the analysis-by-measurement method.

The first method implies that the intuitive, nonverbal knowledge and the experience of an expert musician are translated into performance rules. These rules explicitly describe musically relevant factors. A limitation of this method can be that the rules mainly reflect the musical ideas of specific expert musicians. On the other hand, professional musicians' expertise should possess a certain generality, and in some cases rules produced with the analysis-by-synthesis method have been found to have a general character.

Rules based on an analysis-by-measurement method are derived from measurements of real performances usually recorded on audio CDs or played with MIDI-enabled instruments connected to a computer. Often the data are processed statistically, such that the rules reflect typical rather than individual deviations from a deadpan performance, even though individual deviations may be musically highly relevant.

Many authors have proposed models of automatic music performance. Todd (1992) presented a model of musical expression based on an analysis-by-measurement method. Rule-based

[2] http://www.renconmusic.org.

[3] The Turing test is a proposal for a test of a machine's ability to demonstrate intelligence. Described by Alan Turing in the 1950 paper "Computing Machinery and Intelligence", it proceeds as follows: a human judge engages in a natural language conversation with one human and one machine, each of which try to appear human. All participants are placed in isolated locations. If the judge cannot reliably tell who the machine and the human are, the machine is said to have passed the test.

systems have been proposed by Zanon and Poli (2003), Friberg (1991) and Friberg, Colombo, Frydén, and Sundberg (2000).

Performance systems based on artificial intelligence techniques have been developed too. Widmer (2003) proposed a machine learning based system extracting rules from performances. Ishikawa, Aono, Katayose, and Inokuchi (2000) developed a system for the performance of classical tonal music; a number of performance rules were extracted from recorded performances by using a multiple regression analysis algorithm. Arcos, de Mántaras, and Serra (1998) developed a case-based reasoning system for the synthesis of expressive musical performances of sampled instruments. Delgado, Fajardo, and Molina-Solana (2009) developed a multi-agent approach to music composition and generation.

4. Future challenges

Since the literal synthesis of notes from a score is bland and unappealing, there is an opportunity for learning systems that can automatically produce compelling expressive variations. The problem of synthesizing expressive performance is as exciting as challenging. Music performance is one of the many activities that trained people do very well without knowing exactly how they do it. This is, precisely, one of the main problems to be faced because there is no model that accurately tells us how to perform.

When referring to artistic domains, it is hardly possible to find a 'correct' model whose predictions always correspond with what humans do and what they think is acceptable. We cannot forget that evaluation in these domains is often subjective and heavily-dependent on who is speaking.

Many aspects are involved within expressive performance and it is almost impossible to use them all. Moreover, there are some parameters and dimensions which are commonly considered as non-relevant but that, in fact, might be. Only a portion of the whole problem is tackled by current techniques. One future challenge is to address the problem by using as much dimensions as possible. It could also be possible that some important patterns are hidden and we haven't still discovered them.

Moreover, to obtain very precise data about all those parameters is a challenging problem that cannot still be done in a automatic way. Annotating all this information is a very time-consuming task and requires a lot of effort from several humans. Early systematic investigations in the field have dealt with this problem either by reducing the length of the music (to just some seconds) or by controlling the size of the collections.

Recent approaches try to avoid this task by the use of some statistical learning techniques and by focusing in a more abstract representation of the real notes and their values. Statistical musicology has not historically received much attention, but it is increasing its popularity as problems are getting more and more complex, and the amount of available data grows, even though collect large amount of quantitative data is a really hard task. Temperley (2007) tackles musical perception from a probabilistic perspective in his recent book *Music and Probability*. Apart of proposing a Bayesian network model, the author carries out an interesting survey of works that use statistical tools to solve problems in the *Sound and Music Computing* area.

Despite some successes in computational performance modelling, current models are extremely limited and simplistic regarding the complex phenomenon of musical expression. It remains an intellectual and scientific challenge to probe the limits of formal modelling and rational characterization. Clearly, it is strictly impossible to arrive at complete predictive models of such complex human phenomena. Nevertheless, work towards this goal can advance our understanding and appreciation of the complexity of artistic behaviours. Understanding music performance will require a combination of approaches and disciplines, such as musicology, AI and machine learning, psychology and cognitive science.

For cognitive neuroscience, discovering the mechanisms which govern the understanding of music performance is a first-class problem. Different brain areas are involved in the recognition of different performance features. Knowledge of these can be an important aid to formal modelling and rational characterization of higher order processing, such as the perceptual differentiation between human-like and mechanical performances. Since music making and appreciation is found in all cultures, the results could be extended to the formalization of more general cognitive principles.

Finally but not least, it is the problem of the individuality of each work. Even though there is a huge amount of available data, every song is different from the rest. Hence, it would not be adequate just to apply the way of playing Beethoven's Ninth Symphony to Brahms' Symphonies. A deep study of the work is needed in order to understand the author, the context and the music. One should always keep in mind that artistic performance is far from being predictable.

5. Conclusions

At this point, the question in the beginning of the paper strikes again: can the computer play like a human? This work has tried to offer a comprehensive overview of the current research that is going on in the field of computational expressive music performance. As shown, there is still plenty of room for new research in the area, and the field is currently very active. We have shown the problems been faced as well as the most promising directions for further work.

Studies in music performance have a particular value in our time. The art of performing music is the result of several years of training. At the same time, contemporary information technology offers the possibility of automatic playing of music specially composed for computers or stored in large databases. In such case, the music is typically played exactly as nominally written in the score, thus implicitly ignoring the value of a living performance and its underlying art and diversity.

As seen, research on music performance ranges from studies aimed at understanding expressive performance to attempts at modelling aspects of performance in a formal, quantitative and predictive way. This research can provide expressive tools that traditionally have been hiding in musicians' skill and musical intuition. When explicitly formulated, these tools will give the user the possibility to play music files with different expressive colouring.

Even though we are sceptical about a machine completely replacing a human performer, we are sure that this technology will be available in a not very far future for certain tasks. Scenes like the one in the beginning of this paper will not be science-fiction anymore and it is only a matter of time that they will become commonplace. We have also shown that there are currently some attempts in this direction, like the Rencon contest.

We strongly believe that it is time for computer science to work in the music domain. This research will make a great impact in both the arts and sciences. Not in vain, music is more than an interesting and, somehow, odd domain; it is part of our human essence.

Acknowledgements

This research has been partially supported by the Spanish Ministry of Education and Science under the project TIN2006–15041-C04–01. M. Molina-Solana is also supported by FPU Grant AP2007–02119.

References

Arcos, J. L., de Mántaras, R. L., & Serra, X. (1998). Generating expressive musical performances with SaxEx. *Journal of New Music Research, 27*(3), 194–210.

Bresin, R., & Friberg, A. (2000). Emotional coloring of computer-controlled music performances. *Computer Music Journal, 24*(4), 44–63.

Canazza, S., Poli, G. D., Drioli, C., Rodà, A., & Vidolin, A. (2000). Audio morphing different expressive intentions for multimedia systems. *IEEE MultiMedia, 7*(3), 79–83.

Delgado, M., Fajardo, W., & Molina-Solana, M. (2009). INMAMUSYS: Intelligent multiagent music system. *Expert Systems with Applications, 36*(3), 4574–4580.

De Mántaras, R. L., & Arcos, J. L. (2002). AI and music: From composition to expressive performance. *AI Magazine, 23*(3), 43–57.

Dovey, M. J. (1995). Analysis of Rachmaninoff's piano performances using inductive logic programming. In *Proceedings of the eighth european conference on machine learning (ECML95)* (pp. 279–282). London, UK: Springer-Verlag.

Friberg, A. (1991). Generative rules for music performance: A formal description of a rule system. *Computer Music Journal, 15*(2), 56–71.

Friberg, A. (2006). pDM: An expressive sequencer with real-time control of the KTH music performance rules movements. *Computer Music Journal, 30*(1), 56–71.

Friberg, A., Colombo, V., Frydén, L., & Sundberg, J. (2000). Generating musical performances with director musices. *Computer Music Journal, 24*(3), 23–29.

Gabrielsson, A. (1995). *Music and the mind machine. Expressive intention and performance*. Berlin: Springer [pp. 35–47].

Gabrielsson, A. (2003). Music performance research at the millennium. *The Psychology of Music, 31*(3), 221–272.

Grachten, M., Goebl, W., Flossmann, S., & Widmer, G. (2009). Phase-plane representation and visualization of gestural structure in expressive timing. *Journal New Music Research, 38*(2), 183–195.

Hiraga, R., Bresin, R., Hirata, K., & Katayose, H. (2004). Rencon 2004: Turing test for musical expression. In *Proceedings of the 2004 conference on new interfaces for musical expression (NIME04)* (pp. 120–123). Singapore: National University of Singapore.

Hong, J.-L. (2003). Investigating expressive timing and dynamics in recorded cello. *Psychology of Music, 31*(3), 340–352.

Ishikawa, O., Aono, Y., Katayose, H., & Inokuchi, S. (2000). Extraction of musical performance rules using a modified algorithm of multiple regression analysis. In *Proceedings of the 2000 international computer music conference, San Francisco* (pp. 348–351).

Juslin, P. N. (1997). Perceived emotional expression in synthesized performances of a short melody: Capturing the listeners judgment policy. *Musicae Scientiae, 1*(2), 225–256.

Juslin, P. N., Friberg, A., & Bresin, R. (2002). Toward a computational model of expression in performance: The GERM model. *Musicae Scientiae*(special issue), 63–122.

Langner, J., & Goebl, W. (2003). Visualizing expressive performance in tempo-loudness space. *Computer Music Journal, 27*(4), 69–83.

Lindström, E. (2006). Impact of melodic organization on perceived structure and emotional expression in music. *Musicae Scientiae, 10*, 85–117.

Meyer, L. (1956). *Emotion and meaning in music*. Chicago, IL: University of Chicago Press.

Minsky, M. (1992). *Machine models of music. Music mind and meaning*. Cambridge, MA, USA: MIT Press.

Molina-Solana, M., Arcos, J. L., & Gomez, E. (2008). Using expressive trends for identifying violin performers. In *Proceedings of ninth international conference on music information retrieval (ISMIR2008)* (pp. 495–500).

Palmer, C. (1996). Anatomy of a performance: Sources of musical expression. *Music Perception, 13*(3), 433–453.

Poli, G. D. (2004). Methodologies for expressiveness modelling of and for music performance. *Journal of New Music Research, 33*(3), 189–202.

Ramírez, R., Maestre, E., Pertusa, A., Gómez, E., & Serra, X. (2007). Performance-based interpreter identification in saxophone audio recordings. *IEEE Transactions on Circuits and Systems for Video Technology, 17*(3), 356–364.

Rigg, M. G. (1964). The mood effects of music: A comparison of data from former investigators. *Journal of Psychology, 58*, 427–438.

Sapp, C. (2007). Comparative analysis of multiple musical performances. In *Proceedings of eighth international conference on music information retrieval (ISMIR 2007) Vienna, Austria* (pp. 497–500).

Saunders, C., Hardoon, D., Shawe-Taylor, J., & Widmer, G. (2008). Using string kernels to identify famous performers from their playing style. *Intelligent Data Analysis, 12*(4), 425–440.

Seashore, C. E. (1938). *Psychology of music*. New York: McGraw-Hill.

Sloboda, J. A. (1983). The communication of musical metre in piano performance. *Quarterly Journal of Experimental Psychology, 35*, 377–396.

Stamatatos, E., & Widmer, G. (2005). Automatic identification of music performers with learning ensembles. *Artificial Intelligence, 165*(1), 37–56.

Sundberg, J. (2000). Emotive transforms. *Phonetica, 57*, 95–112.

Temperley, D. (2007). *Music and probability*. The MIT Press.

Todd, N. P. (1989). A computational model of rubato. *Contemporary Music Review, 3*(1), 69–88.

Todd, N. P. (1992). The dynamics of dynamics: A model of musical expression. *Journal of the Acoustical Society of America, 91*(6), 3540–3550.

Widmer, G. (2003). Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries. *Artificial Intelligence, 146*(2), 129–148.

Widmer, G., Dixon, S., Goebl, W., Pampalk, E., & Tobudic, A. (2003). In search of the Horowitz factor. *AI Magazine, 24*(3), 111–130.

Widmer, G., & Goebl, W. (2004). Computational models of expressive music performance: The state of the art. *Journal of New Music Research, 33*(3), 203–216.

Zanon, P., & Poli, G. D. (2003). Estimation of parameters in rule systems for expressive rendering in musical performance. *Computer Music Journal, 27*(1), 29–46.

INMAMUSYS: Intelligent multiagent music system

Delgado, M., Fajardo, W. & Molina-Solana, M. (2009), "*INMAMUSYS: Intelligent multiagent music system*", Expert Systems with Applications. Vol. 36(3), pp. 4574-4580, DOI: 10.1016/j.eswa.2008.05.028

- Status: **Published**
- Impact Factor (JCR 2009): 2.908
- Subject category:
  - COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE (15/103 Q1)
  - ENGINEERING, ELECTRICAL & ELECTRONIC (16/246 Q1)
  - OPERATIONS RESEARCH & MANAGEMENT SCIENCE (3/73 Q1)

Expert Systems with Applications 36 (2009) 4574–4580

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Inmamusys: Intelligent multiagent music system

Miguel Delgado, Waldo Fajardo, Miguel Molina-Solana *

Department of Computer Science and Artificial Intelligence, Universidad de Granada, 18071 Granada, Spain
E.T.S.I. Informática y de Telecomunicación, C\Periodista Daniel Saucedo Aranda, 18071 Granada, Spain

ARTICLE INFO

Keywords:
Computer music
Music composition
Multiagent systems
Rule-base system

ABSTRACT

Music generation is a complex task even for human beings. This paper describes a two level competitive/collaborative multiagent approach for autonomous, non-deterministic, computer music composition. Our aim is to build a high modular system that composes music on its own by using Experts Systems technology and rule-based systems principles. To do that, rules issued from musical knowledge are used and emotional inputs from the users are introduced. In fact, users are not allowed to directly control the composition process. Two main goals are sought after: investigating relationships between computers and emotions and how the latter can be represented into the former, and developing a framework for music composition that can be useful for future experiments. The system has been successfully tested by asking several people to match compositions with suggested emotions.

1. Introduction

Surely music is one of the most difficult human disciplines. It requires creativity, specific knowledge and eventually, some manual abilities. It is commonly said that music is something humans do, but something we do not understand. There are several mechanisms involved in music and unfortunately, many of them are still unknown. Others are so complex that we cannot manage them with actual computer tools.

On the other hand, Computer Science has suffered a huge evolution in just a small period of time. However, despite all the great applications and problems solved during the last decades, there are many problems that computers are unable to deal with nowadays.

Among others, we can point fuzzy representation of abstract concepts and, in general, dealing with feelings and emotions. In fact, cognitive processes involving reasoning, knowledge and experience are hardly represented in a computer and are actual hotspots for Artificial Intelligence.

Music is a good example of applied AI related with those topics. It has been demonstrated that music composition is a hard task even for humans, so using machines appears as a very interesting and fascinating area of research. AI techniques are going to be applied to the musical domain with the aim of understanding human musical abilities. Because there are so many challenges to deal with, a bottom-up approximation is required for solving all of them in a modular way.

The paper is organized as follows. In Section 2, we examine some previous works done in the field. Section 3 deals with the architecture of the proposed system. In Section 4, we discuss design and implementation of system components. Section 5 presents an evaluation method and summarizes some users' impressions about the output; and in the last section, future work to be done is presented and discussed.

2. Background

Since first computers were developed many people have tried to apply them to musical tasks.

There are two main classes in which computer music projects could be classified: analysis and composition. The first one consists on extracting information from the music itself (or the associated data) in order to learn some rules, or go to a model that describes the concrete examples. Because this is not the main field of this paper, so we are not going to go further. However, Anagnostopoulou and Westermann (1997), and Balaban (1996) could be reviewed for more information.

Composition is about generating new music from the rules. In fact, is doing the process in the other way: 'from rules to music' instead of 'from music to rules'. According to Pearce, Meredith, and Wiggins (2002), the final objective of most of the compositional prototypes is to demonstrate that standard musical techniques could be handled by computer programming, and also to validate generative music theories.

Initial approaches in algorithmic composition consist on randomly selecting notes (mainly pitch and rhythm) with some constraints in order to generate compositions. This vision produced limited results but was a great point of departure for later works.

Experts system has been widely used to compose music. Rules concerning pitch, duration and volume have tried to apprise the

* Corresponding author.
E-mail address: miguelmolina@ugr.es (M. Molina-Solana).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.05.028

knowledge involved in human composition. Friberg (1991) introduced an early example of rule system. Many works on computer music generation are based on the approach that the composition rules are specified by the composer. So that, an expert is needed in order to give all the knowledge. The problem here is that Music Theory is not as formal as it should be to be easily represented in a computer. However, Lerdahl and Jackendoff (1983) proposed an interesting model to generate music using grammars.

Another approach to compose music with computers is to use genetic algorithms. Genetics have been successfully applied to several problems with difficulty in defining the solution process, or where searching a huge solution space is needed. Composition falls into this class of tasks, and some works in this direction can be found in Marques, Oliveira, Vieira, and Rosa (2000) and Miranda (2004).

Although many advances have been done during last decades in Computer Music in general and in algorithmic composition in particular, it is true that the greatest moment was when Cope (1996) presented EMI Project.

Many prototypes, throughout the years, have demonstrated that computers algorithms cannot be compared with human minds. Machines just produce very simple compositions in a quite mechanical manner.

Computer programs that perform music with proper stylistic considerations (like a human expert would) are very scarce and only work for a few examples and in very specific domains may be found. The main problem is that algorithms rarely deal with feelings. And music without emotions is unworthy and not natural.

Minsky (1991) expressed his conviction that the unique way for creating a machine whose creations transmit something to listeners, should start by simulating emotions in computers. Meyer (1956) addressed this topic (expression in computers) from a musicological point of view, while Narmour (1990) proposed a cognitive model for understanding melodies.

This topic is a current hotspot in computer music, and many groups are working on it right now. Kiendl, Kiseliova, and Rambinintsoa (2006), López de Mántaras and Arcos (2002), Qijun Zhang and Miranda (2006), Widmer (2001) and Wieczorkowska (2004) can be mentioned as some relevant examples. For further information about composing music with computers, Cope (2005) and Miranda (2001) works should be reviewed.

In recent years, agent paradigm has become quite common. Almost everybody is using it for everything. The reason is that agents are a very powerful way of implementing distributed AI. Indeed, they can be combined with others tools, such as rule-based systems, case-based systems or searching algorithms. Actually, Agent Theory just defines interactions between agents, not how they are internally built. Because of that, we can have a multiagent system where agents can be implemented with just an algorithm, using CBR techniques, rules, genetics... Interesting work related with multiagent systems can be found in Todd and Werner (1999), Wulfhorst, Nakayama, and Vicari (2003).

3. Architecture

The architecture we propose in this paper is a two-layer multiagent system. The current application of multiagent systems in real-time environments is an area of increasing interest. In general, multiagent systems are an appropriate approach for solving inherently distributed problems, whereby clearly different and independent processes can be distinguished.

The first level is the competitive one, where agents (called *composers*) compete among themselves to be the one chosen for composing. This layer allows us to make an initial separation between composition styles. We think it does not make any sense to have a one-for-all agent, so a collection of simple agents specialized in some task is proposed. Each composer announces its abilities, and the system, according with user inputs, selects the composer that better fits. See Fig. 1.

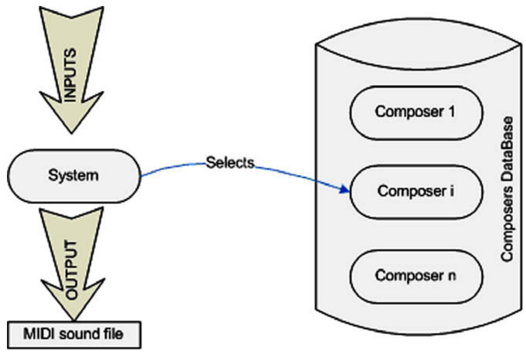


Fig. 1. First level architecture.

This agent chooses, between other parameters, rhythm, number of voices, and instruments to be used in the compositions. However, the selected composer only acts as a director, being useless on its own. Going further with the modularization principle, the composer agent finally asks some others agents from the second level for their collaboration in order to get a solution.[1]

This second level contains auxiliary agents that collaborate between them, so this layer should be understood as a collaborative level. We call these agents *voice generators*, because mainly that is their job. See Fig. 2.

In general, there are just three voices in normal compositions: melody, accompaniment and harmony. However, the system presents no limitations in relationships between agents. A composer agent can employ as many voices as desired, and in the other hand, several composers could use the same voice generator.

When some intelligence is needed, agents are to be designed as intelligent rule-base systems. Otherwise, they just implement a simple algorithm. At the end, we have a big set of simple agents that together, manage to find a whole solution for the problem.

4. Design

INMAMUSIC current prototype has several composers that generate music in different ways: from just a random composer to more elaborate ones where aesthetic principles are mainly searched.

An important design principle that is going to guide our system is creating a tool that can be used by everyone. Most prototypes present a very complex interface that only experts can understand, and even for them, it is very annoying to fill a huge amount of data in order to get a composition. In fact, when people speak about music, they do not usually use technical terms such as granularity, rhythm, or tonality; they use emotions, feelings and abstract concepts. Actually, they are not speaking in low level, but in a high level, where technical parameters do not exist. Keeping that in mind, our aim is the users ought to be faced with a friendly interface with only a few questions; and not really difficult ones, just questions about the wished music style. In other words, a high level music interface is to be introduced.

[1] Solution in this context means just a composition that the system offers as output, without evaluating its quality.

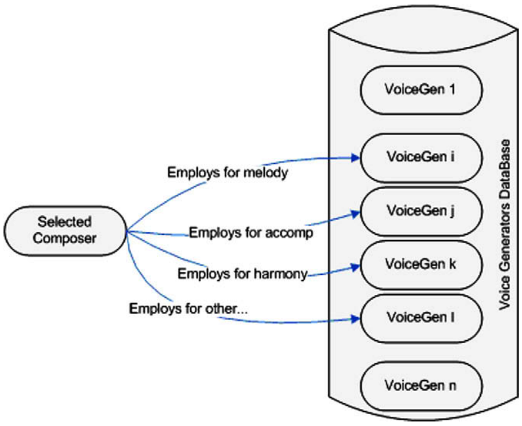


Fig. 2. Second level architecture.

In fact, we are integrating into the system itself the knowledge about how to use tonality, rhythm and instruments in order to get, for instance, a sad music.

Until now, this task has been left in the hands of humans. We propose to go further providing the machine with the suitable resources it should use to compose in a certain way. The human does not have to worry about dozens of parameters and how they should be combined to get a certain result.

Java has been chosen as programming language due to its object oriented style and ease of use. jMusic library is used for music representation and processing. This library is described by Brown and Sorensen (2000).

4.1. Knowledge representation

We have made use of several ways to represent information. Firstly, because most agents are designed as rule-base systems, we need to represent some information through IF-THEN rules. They are implemented in XML in order to achieve a universal way of representation. Fig. 3 shows the XML schema for rules in the DTD language. Basically, a knowledge base is a set of several rules. A rule has two parts: IF and THEN. IF part consists of some attribute-value pairs; and THEN has a list of consequents, that can be viewed as actions to be done. We have decided to employ several rules subsets in order to respect the modularity, and to make the inference engine's job easier. Higher performance is also reached with this approach.

To write the code, we use the representation that jMusic API offers us. This is an object oriented one, very close to normal Western classical music representation. There are manuals and several examples in jMusic website (http://jmusic.ci.qut.edu.au). As agents are developed for a concrete musical style or concrete task, there is obviously a lot of heuristic information about how to compose coded in each agent. Agents are implemented with certain parameters (that only an expert knows) in order to accomplish its goals. This kind of knowledge should be minimized and moved to a unique and standard repository, in order to obtain an independent knowledge database. However, many times it is not easy to code procedural information in a data format, so we should be aware that much knowledge about the topic is into the composition process itself.

```
<!ELEMENT knowledgebase (rule)+>
<!ELEMENT rule (if, then)>
<!ELEMENT if (atrib,value)+>
<!ELEMENT atrib ANY>
<!ELEMENT value ANY>
<!ELEMENT then (consec+)>
<!ELEMENT consec ANY>
```

Fig. 3. XML schema in DTD language for representing rules in the system.

System output is a sound file in MIDI format. This representation is preferred rather than others audio formats such as wav or mp3, because it is easier for symbolic manipulation. Even more, MIDI could be easily converted into the others two (if needed), but the opposite is not true. Eventually, a music sheet could be generated if needed.

4.2. Composition agents

Composer agents, as we have previously said, are located in the first layer of the architecture. They compete with others in order to be elected for composing. Basically, all composers are in charge of defining the number of voices the composition will have, instruments to be used, the measure and tonality. They also indicate which second level agents should be used for generating voices. In other words, composers act as directors. In current system there are four implemented composers: Muzak, Dark, Scales and Random.

Muzak Composer generates music in a Muzak[2] way. This kind of music, also called ambient or elevator music, could be described as a soft and quiet one. Brian Eno wrote "Ambient Music must be able to accommodate many levels of listening attention without enforcing one in particular; it must be as ignorable as it is interesting". From this point of view, ambient music does not need to be very elaborate or complex. Its only requisite is to be pleasant and agreeable to the human ear.

Dark Composer aim is to compose music that provokes fear in listeners. To do that a set of suitable resources is used. Dissonances and diminished fifth interval (known as tritone) are heavily employed by this agent. This interval is called *diabolus in musica* (the Devil in music) and has been historically avoided. Great distances between voices and deep bass chords are also useful resources that are employed.

Scales Composer is quite simple. It just produces some ascendant and descendent scales in random tonalities and modes. The mixing of several voices doing the same in different moments, velocities and tones produces interesting effects.

Finally, we introduce the Random Composer with its two variants: the "differential", and the "independent". In the first one, intervals are randomly generated, so a note pitch depends on previous note. The other option is to randomly generate pitches, so a note is independent of the rest. Durations are also generated with a random generator. As the reader can imagine, a composer acting this way produces chaotic compositions without any internal coherence. So that, chaos is the tag that better defines the music this composer generates.

4.3. Voice generators

Voice Generators are in charge of producing sounds with different volumes, durations and intonations for every melodic line.

Many kinds of voice generators could be implemented, but the main ones fall into any of the following classes: Melody, harmony, accompaniment and drums. However, the system architecture is flexible enough to allow any other desired voice. In fact, the

[2] MUZAK is the name of a company specialized in this kind of music. People often refer to ambient music in this way.

architecture does not care about the semantic meaning of a certain melodic line.

4.3.1. Harmony generator

Harmony generator makes use of a set of rules that indicates which tonal movements are allowed. In this way, chord progressions can be generated. With this, we have the skeleton of compositions. We can understand this as a Markov chain model. In Fig. 4 we can see the set of rules in the current implementation, that correspond with Fux's counterpoint rules (Piston, 1947).

4.3.2. Melody generator

Melody generators produce the melody of compositions. Current design uses a big set of motives which are one measure long. A motive is randomly selected in each measure and put into the melody voice. However, preliminary tests reveal that acting this way, compositions will not be coherent, in the sense that they will not seem as a unique entity, but rather, like little pieces stuck together.

The trick here consists in to randomly select (from the whole set) a subset of motives to be used in each execution. To have a small number of motives for each execution makes the composition appear as a whole, giving the impression of melodic phrases. In fact, there is no structure at all to be followed, but the repetition over and over again of similar motives induces the listener to think so. Employing this simple trick the output seems to be organized in phrases (but as seen there is no more than random selection) and the impression is quite realistic.

Moreover, due to the fact that the subset is randomly selected, it is different for each execution. This is an important point because it produces a great variety between two compositions, even thought when they are generated with same inputs.

```
IF actual_degree = I   THEN next_degree = I
IF actual_degree = I   THEN next_degree = II
IF actual_degree = I   THEN next_degree = III
IF actual_degree = I   THEN next_degree = IV
IF actual_degree = I   THEN next_degree = V
IF actual_degree = I   THEN next_degree = VI
IF actual_degree = I   THEN next_degree = VII
IF actual_degree = II  THEN next_degree = II
IF actual_degree = II  THEN next_degree = IV
IF actual_degree = II  THEN next_degree = V
IF actual_degree = II  THEN next_degree = VI
IF actual_degree = II  THEN next_degree = VII
IF actual_degree = III THEN next_degree = II
IF actual_degree = III THEN next_degree = IV
IF actual_degree = III THEN next_degree = V
IF actual_degree = III THEN next_degree = VI
IF actual_degree = IV  THEN next_degree = I
IF actual_degree = IV  THEN next_degree = II
IF actual_degree = IV  THEN next_degree = III
IF actual_degree = IV  THEN next_degree = V
IF actual_degree = IV  THEN next_degree = VI
IF actual_degree = IV  THEN next_degree = VII
IF actual_degree = V   THEN next_degree = I
IF actual_degree = V   THEN next_degree = III
IF actual_degree = V   THEN next_degree = IV
IF actual_degree = V   THEN next_degree = VI
IF actual_degree = VI  THEN next_degree = II
IF actual_degree = VI  THEN next_degree = III
IF actual_degree = VI  THEN next_degree = IV
IF actual_degree = VI  THEN next_degree = V
IF actual_degree = VII THEN next_degree = I
IF actual_degree = VII THEN next_degree = III
IF actual_degree = VII THEN next_degree = V
IF actual_degree = VII THEN next_degree = VI
```

Fig. 4. Rules used by harmony generator to generate chord progressions.

4.3.3. Drums generator

Drums generators are in charge of generating a continuous rhythm. There are several composers that can be classified within this class. They implemented different rhythms: some offers more density, others are lighter; ones use various sounds (drum, snare, cymbal, hat...), others just use a simple drum. At the end, it is the first level agent (the composer) the one in charge of selecting the appropriate Drum Generator (or even none of them) for each occasion.

4.4. User interface

Our main aim was to develop an easy and simple interface that anyone could use without problems.

Before, we talked about the purpose of implementing an easy-to-use system. The objective here is presenting to the users an interface that anyone can use, even if they are not experts. To do that, we need to include within the system all the information related to technical parameters. This is a great deal because the matching between emotions and music parameters is not direct, and to define parameters themselves is definitively not a trivial task.

The interface (see Fig. 5) is very dynamic and not fixed because its contents depend on the composers implemented in the system and the capabilities they announced. The form only contains terms that agents have declared in some rule. With this, we avoid unexpected inputs and assure the system will always be able to give us a valid output. Also, it lets us add, replace, or modify agents when any problems appear, making changes easier.

In addition, we have added the possibility of selecting which instruments should be use in each voice. However, if we do not specify an instrument, the system selects one from those it thinks is a better fit. Not all instrument combinations sound good, and even if they do, it is possible that they are not compatible with the style of the melody. These options exist in order to demonstrate the complexity of the knowledge involved. Choosing rhythm, instruments, tonality or granularity are not trivial tasks, because they should be considered all together, and related with some other concepts. By allowing selection of instruments, users can test if they are able to find combinations that really make great compositions, beating system choices.

4.5. The composition process

In this subsection, the composition process of Muzak Composer is described to illustrate how a composer runs because the process is quite similar for the rest of composers.

When Muzak Composer is selected by the Manager Agent, it firstly decides the tempo of the composition attending to user inputs. The mapping between linguistics tags and exact bpm (beats per minute) is done by using a normal random number generator with given mean and variance. The means values for each tag have been statistically obtained from both a set of classical sheet music and experts' feedback.

As said before, composer agents are also in charge of selecting the number of voices to be used, and the second level agents are responsible for their generation.

To begin with, the harmony generator is executed, producing a chord progression which is the skeleton of the composition.

After that, it is the melody generator's turn. For this agent and in the current implementation, a motive database is decided to be used (as commented before), as well as employing one of its elements in each measure. Even thought there is no kind of structure

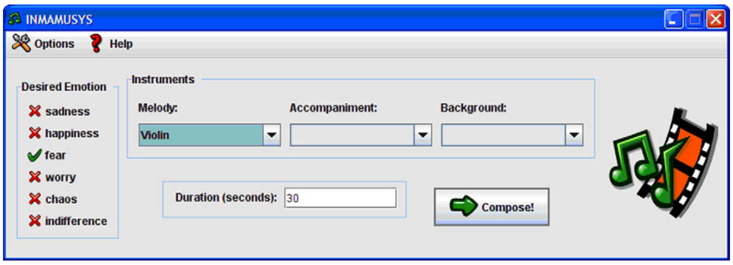


Fig. 5. INMAMUSYS' user interface.

of phrases, or a grammar to be followed, the global composition seems to have an internal structure. This effect is suggested by frequent motive repetitions.

Next step is to execute the accompaniment generator. Goal here is to complete the harmonization of the composition. Accompaniment generator introduces some new notes and completes chords. It could just be a note every measure, an arpeggio or perhaps something more elaborate.

It is important to indicate that melody and accompaniment are generated always in C major. This is not a handicap and greatly facilitates the task. However, it is necessary to transpose them to the correct position in each measure, according to the tonality and the current grade. Also, the note in the first time of each measure must be accentuated. It is up to voice generators to assure that.

5. Experiments and evaluation

The evaluation of any musical work is a complex task and often comes down to individual subjective opinion. It depends not only on formal aspects but also on some stylistics ones. Because of that, it is hard to empirically evaluate music compositions, and therefore it is difficult to evaluate the effectiveness of a computer music composition system.

Many metrics could be developed, but all of them will fail as at the end, music (understanding the term as much more than just a chain of sounds) cannot be reduced to a number.

Due to this difficulty, many authors will typically conclude their papers with a vague comment such as "compositions generated by the system are quite impressive and very promising" or "sometimes, melody seems to be a bit simple and unelaborated; but many, results are very human like". However, this is unsatisfactory for two reasons: first, evaluating the music produced by the system reveals little about its utility as a compositional tool; and second, qualitative and subjective evaluation by the designers of the system reveals little about the value of the tool to other composers (Pearce et al., 2002).

This author affirms that there is an historical malaise in adopting suitable evaluation procedures for judging the degree to which the aims have been satisfied. As we agree with this assertion, we have also proposed an evaluation method to assess compositions in the same way that they are usually appraised: through audience

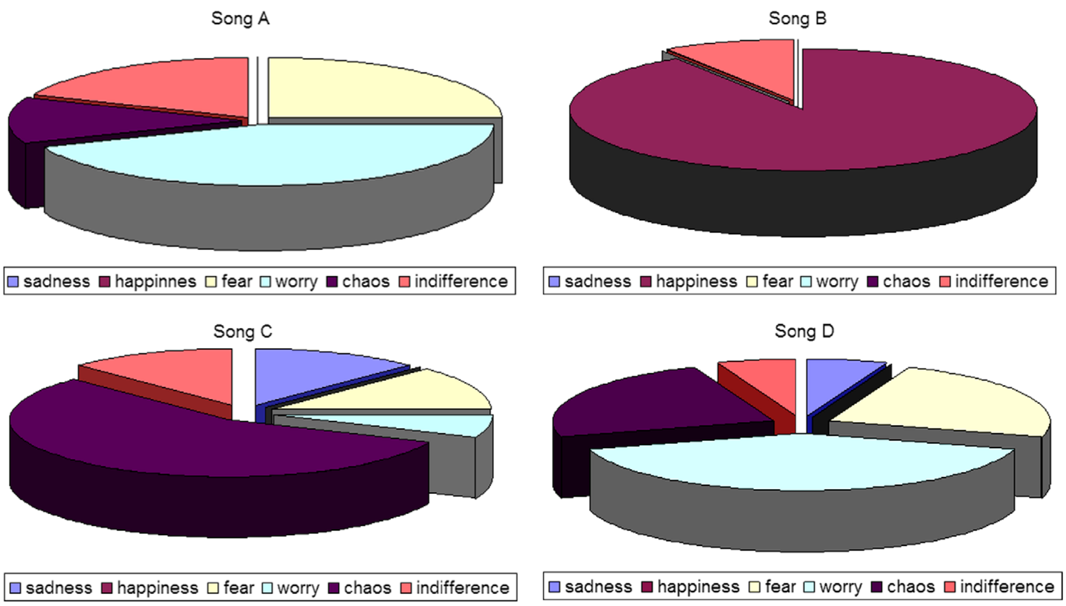


Fig. 6. Results from the test: Feelings that the four tested songs provoked in listeners.

reactions and critical reviews. So that, a short test has been design to carry out that task.

5.1. Experiments

To begin with, we asked some people for listening and evaluating some examples generated by INMAMUSYS. Most of them could not believe that a machine was the real author. For us, this is an important point because capturing the essence of a human's composition method is a great deal. This fact makes us also think that our system could have passed, in some sense, the Turing test.

The second part of the proposed evaluation method was more formal and consists in testing what emotions were induced on listeners when listening to some compositions. The objective was to probe whether compositions generated by the system provokes in listeners emotions such as those that guided the composition process and were given as inputs to the system. This test will give us a measure of how the system is able to compose music that successfully matches user emotional requests.

Four compositions were generated with different input values. Songs A, B, C and D are respectively generated by using the inputs worry, happiness, chaos and worry again (this input is used twice). System mapped these emotions into the use of Dark, Muzak, Random and Dark composers. The four examples were presented to the listeners without giving them information about the origin of the songs, and the listeners were asked to select emotions that better fit what they listen. Users could tick as much tags as they want from the list: sadness, happiness, fear, worry, chaos and indifference. Twenty people, involving a huge range in musical expertise, participate in our experiment by answering the questions.

Results can be found in Fig. 6. We were really surprised by them because we did not expect such a clear confirmation, even more when the emotion-composer matching mechanism is quite primitive. In all the four cases, the most repeated option is the right one (the one that was given as input to the system). It is true that the list of potential options is small, (we would like to make it bigger the next time), but it is also clear that differences between some tags are very small, so that it would have been easy to get some wrong answer.

It is also noticeable, that many people (12 out of 20) point out that the first song and the last one were very similar. They are definitively different but these answers makes us think that *Dark composer* outputs do not cover the whole area at the solution space that we supposed, but just a little zone. We think this should be corrected in the future, because is important to accomplish the objective that all the compositions from the same inputs fall in the same area, but it is also crucial to maximize this area, not just focusing in a point.

To conclude, we would like to comment that an equilibrated system has been developed: on the one hand, there is a great variety between different executions; on the other, we have managed that each composition sounds as a whole not like several pieces stuck together. As said, the whole solution space for a tag (e.g. fear) is not completely cover, but at least we can say that almost all the system outputs are classified (by humans) under the right tag.

6. Conclusions and future work

Composing music is a very complex process that involves many disciplines and tasks. In this paper we have presented a new approach for composing music using computers. Because there are so many challenges to deal with, a bottom-up approximation is required for solving all of them in a modular way. We are interested in building a framework for successfully composing music that provokes some feelings in listeners.

In order to achieve this goal, we have proposed here a two-level architecture that successfully deals with the complex problem of music composition, so that, a huge problem can be broken into smaller tasks. This approach makes use of several agents and rule-based systems. It also permits that users were provided by an easy-to-use interface that hides all the complexity of music composition. Even more, inputs for this interface are emotional inputs from the users, so that we are able to address the problem of music expressiveness.

Several design decisions that affect the system were taken and we have explained them. We have focused in knowledge representation because it is a main topic. In that section we also describe different agents, as well as the role they play in the whole system. At the moment, just four kinds of composers have been developed: Dark, Muzak, Scales and Random.

Finally, results obtained after evaluating the actual system are showed. This evaluation has been carried out with an experiment and in a formal way. Even though the system is in a quite early stage of development, results are promising enough to encourage us to continue working with this framework.

6.1. Future work

The current prototype is quite rigid in the way it deals with user inputs. By now, we are just using classic rule matching for selecting agents, and that is a poor approach for dealing with fuzzy concepts, such as emotions and musical terms. So that, including fuzzy logic in the inference system is probably the first modification to be done. This change would require major modifications in the inference engine, as well as the definition of fuzzy domains and linguistic tags. However, we think it would be worth the effort.

Secondly, we would like to develop more composers in order to get a bigger collection of these kinds of agents. Our aim is to obtain as much diversity as possible not only to compare different algorithms and compositional mechanisms, but for implementing and testing new ideas. These ideas could be related with new theories of musical styles or cognitive processes. Even more, as the set of composers growths, we will be able to reproduce a wider range of human emotions.

Another interesting project would be to develop a module to automatically obtain composition rules. In the current prototype, this knowledge is given by humans, and coded into the agents. It would be very powerful if the system could analyze a music sheet, then to extract some rules, and thus to compose according to them. The system will win a great flexibility with this ability. Not in vain, this one is the key idea in Cope's system, and an interesting hotspot in actual data mining. Doing this, the system would be quite complete, in the sense that it would include both composition and analysis tasks.

As seen, there is plenty of room for researching in the computer music area in general and in composition in particular. At our department, we feel computer music is a great opportunity for modern Artificial Intelligence. We have developed this project with the aim of it being used as a framework for future experiments and works.

We also believe that Expert Systems and Agent Theory have many things to say in this field. Composing music is a huge problem that should be divided into several tasks, and it definitively needs intelligence to be done.

Acknowledgement

This research has been partially supported by the Spanish Ministry of Education and Science under the project TIN 2006-15041-C04-01.

References

Anagnostopoulou, C., & Westermann, G. (1997). Classification in music: A computational model for paradigmatic analysis. In *Proceedings of the International Computer Music Conference* (pp. 125–128).

Balaban, M. (1996). The music structures approach in knowledge representation for music processing. *Computer Music Journal, 20*(2), 96–111.

Brown, A., & Sorensen, A. (2000). Introducing jMusic. In *Proceedings Australasian computer music conference* (pp. 68–76).

Cope, D. (2005). *Computers models of musical creativity*. The MIT Press.

Cope, D. (1996). *Experiments in musical intelligence*. Madison, WI: A-R Editions.

Friberg, A. (1991). Generative rules for music performance: A formal description of a rule system. *Computer Music Journal, 15*, 56–71.

Kiendl, H., Kiseliova, T., & Rambinintsoa, Y. (2006). Use of fuzzy strategies for interpretation of music. *Journal of Multiple-Valued Logic and Soft Computing, 12*(1), 149–169.

Lerdahl, F., & Jackendoff, R. (1983). *A generative theory of tonal music*. Cambridge, Mass: MIT Press.

López de Mántaras, R., & Arcos, J. L. (2002). AI and music. From composition to expressive performance. *AI Magazine*(Fall), 43–58.

Marques, M., Oliveira, V., Vieira, S., & Rosa, A.C. (2000). Music Composition using genetic evolutionary. In *Proceedings of the 2000 congress on evolutionary computation* (pp. 714–719).

Meyer, L. B. (1956). *Emotion and meaning in music*. Chicago, IL: University of Chicago Press.

Minsky, M. (1991). *A conversation with Marvin Minsky*. *Understanding musical activities: Readings in AI and music*. Menlo Park: AAAI Press.

Miranda, E. R. (2001). *Composing music with computers*. Oxford, UK: Focal Press.

Miranda, E. R. (2004). At the crossroads of evolutionary computation and music: Self-programming synthesizers, swarm orchestras and the origins of melody. *Evolutionary Computation, 12*(2), 137–158.

Narmour, E. (1990). *The analysis and cognition of basic melodic structures: The implication-realization model*. Chicago: University of Chicago Press.

Pearce, M. T., Meredith, D., & Wiggins, G. A. (2002). Motivations and methodologies for automation of the compositional process. *Musicae Scientiae, 6*(2), 119–147.

Piston, W. (1947). *Counterpoint*. W.W. Norton.

Qijun Zhang, & Miranda, E. R. (2006). Towards an Evolution Model of Expressive Music Performance. In *Sixth international conference on intelligent systems design and applications, ISDA'06* (pp. 1189–1194).

Todd, P. M., & Werner, G. M. (1999). *Frankensteinian methods for evolutionary music composition*. *Musical networks: Parallel distributed perception and performance*. Cambridge, MA: MIT Press/Bradford Books.

Wieczorkowska, A. (2004). Towards extracting emotions from music. *Intelligent media technology for communicative intelligence, IMTCI*, 228–238.

Widmer, G. (2001). The musical expression project: A challenge for machine learning and knowledge discovery. In *Proceedings of the 12th European conference on machine learning, EMCL 2001* (pp. 603–614).

Wulfhorst, D., Nakayama, L., & Vicari, R.M. (2003). A multiagent approach for musical interactive systems. In *Proceedings of the second international joint conference on autonomous agents and multiagent systems* (pp. 584–591).

Mining transposed motifs in music

Jiménez, A., Molina-Solana, M., Berzal, F. & Fajardo, W. (2011), "*Mining transposed motifs in music*", Journal of Intelligent Information Systems. Vol. 36(1), pp. 99-115, DOI: 10.1007/s10844-010-0122-7

- Status: **Published**
- Impact Factor (JCR 2010): 0.875
- Subject category:
  - COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE (76/108 Q3)
  - COMPUTER SCIENCE, INFORMATION SYSTEMS (82/128 Q3)

J Intell Inf Syst (2011) 36:99–115
DOI 10.1007/s10844-010-0122-7

Mining transposed motifs in music

Aída Jiménez · Miguel Molina-Solana · Fernando Berzal · Waldo Fajardo

Received: 27 November 2009 / Revised: 23 February 2010 / Accepted: 21 March 2010 /
Published online: 13 April 2010

Abstract The discovery of frequent musical patterns (motifs) is a relevant problem in musicology. This paper introduces an unsupervised algorithm to address this problem in symbolically-represented musical melodies. Our algorithm is able to identify transposed patterns including exact matchings, i.e., null transpositions. We have tested our algorithm on a corpus of songs and the results suggest that our approach is promising, specially when dealing with songs that include non-exact repetitions.

Keywords Musical mining · Motifs · Frequent pattern mining

1 Introduction

The discovery of frequent musical patterns (motifs) is a relevant problem in musicology. In music, we can find several entities that can be repeated such as notes, intervals, rhythms, and harmonic progressions. In other words, music can be seen as a string of musical entities such as notes or chords on which pattern recognition techniques can be applied.

We can define a music motif as *the smallest meaningful melody element*. As a rule, motifs are groups of notes no longer than one measure. In human speech, a motif is a word. In the same way that sentences consist of words, motifs form musical phrases. A melody is formed by several main motifs, which are repeated, developed, and opposed one against another within the melody evolution.

A. Jiménez ( ) · M. Molina-Solana · F. Berzal · W. Fajardo
Centro de Investigación en Tecnologías de la Información y las Comunicaciones,
University of Granada, Granada, Spain
e-mail: aidajm@ugr.es

M. Molina-Solana
e-mail: miguelmolina@ugr.es

F. Berzal
e-mail: berzal@acm.org

W. Fajardo
e-mail: aragorn@ugr.es

When analyzing a music work, musicians carry out a deep analysis of the musical material. This analysis includes motif extraction as a basic task. Musician studies include contextual information (such as the author, the aim, or the period) but also morphological data from the music itself. Looking for the motifs that build the whole work is the first step that a musician takes when faced with a music sheet.

Audio-thumbnailing (i.e., summarizing or abstracting) is another interesting application in the musical domain that is related to motif extraction. It provides the user with a brief excerpt of a song that (ideally) contains the main features of the work. Before hearing or purchasing a whole song, it would be useful to hear a representative thumbnail of the whole work. This technique is also important for indexing large datasets of songs, which can be browsed more quickly and searched more efficiently if indexed by those small patterns instead of being indexed by the whole song.

One of the most fundamental ways to classify MIR methods is to divide them into those that process audio signals using signal processing methods and those that process symbolic representations. We have decided to work with a symbolic representation instead of an audio one because it is closer to the original sheet of music. In other words, the main difficulty with audio representation is that the transformation from audio signals to symbolic data is far from being accurate. This fact makes the pattern recognition problem much more difficult and it requires completely new techniques to deal with signals.

Using the algorithm we present in this paper, we are able to find frequent melodic and rhythmical patterns in music starting from the *MusicXML* representation of the song (www.wikifonia.org). We first transform this symbolic representation into a sequence of notes. These notes are defined at their lowest level (i.e., pitch and duration) and in an absolute, not relative, way.

According to the above considerations, we have developed a TreeMiner-based (Zaki 2005b) algorithm to discover frequent subsequences in music files. Our algorithm is able to identify sequences even when they are transposed. It can be used to find common motifs in several songs and also find repetitions within a song. In this paper, we present its application to the discovery of long motifs that are repeated within a single song. Our hypothesis is that those patterns probably correspond to the chorus or the more significant part of the song.

Our paper is an extended version of a paper presented at the ISMIS'09 conference (Berzal et al. 2009) and is organized as follows. In Section 2, we provide some background on musical data mining and introduce some relevant terms. Section 3 formally defines our sequence pattern mining problem and describes the algorithm we have devised to solve it. In Section 4, we explain the way our algorithm works by means of a particular example. Some experimental results are presented in Section 5, whereas in Section 6 we draw some conclusions.

2 Background

Although it is almost impossible to be exhaustive in analyzing the state of the art in musical pattern identification, we survey the most relevant works in this field in Section 2.1. As our approach is based on sequences, we introduce some standard terms and review some sequence mining algorithms proposed in the literature in Section 2.2.

2.1 Data mining in music

Pattern processing techniques have been applied to musical strings. A complete overview can be found, for instance, in the paper by Cambouropoulos et al. (2001). Those algorithms can be divided into those that deal with audio signals (using signal processing methods) and those that use symbolic representations.

2.1.1 Dealing with audio signals

There are several researchers that have addressed the problem of pattern induction in an acoustic signal. For instance, Aucouturier and Sandler (2002) proposed an algorithm to find repeated patterns in an acoustic signal by focusing on timbre; whereas Chu and Logan (2002) proposed a method to find the most representative pattern in a song using Mel-spectral features.

Recently, some works have gone further in this direction by trying to identify the sectional form of a musical piece from an acoustic signal. For example, Paulus and Klapuri (2009) address this task using a probabilistic fitness measure based on three acoustic features; whereas Levy and Sandler (2008) use clustering methods to extract this sectional structure.

Solving the problem of identifying the structure of a musical piece is key for audio-thumbnailing (i.e., finding a short and representative sample of a song). Zhang and Samadani (2007) addressed this problem by detecting paragraphs in the song with repeated melody in a first step and then identifying vocal portions in the song. With such information, the structure of the song is derived. Another approach, by Bartsch and Wakefield (2005), developed a chroma-based system that searches for structural redundancy within a given song with the aim of identifying something like a chorus or refrain.

2.1.2 Using symbolic representations

There are certain similarities in the use of text and musical data which also allow the application of text mining methods to process musical data. Both have a hierarchical structure and the relative order among the elements is of importance. For that reason, researchers have proposed many different meaningful ways of representing a piece of music as a string, but all of them use either event strings (where each symbol represents an event) or interval strings (where each symbol represents the transformation between events).

Most of the proposed techniques start from a symbolic transcription of music. For example, Hsu et al. (1998) used a dynamic programming technique to find repeating factors in strings representing monophonic melodies; whereas Rolland (1998) recursively computed the distances between large patterns from the distances between smaller patterns. Meredith et al. (2002) proposed a geometric approach to repetition discovery in which the music is represented as a multidimensional dataset. Pienimäki (2002) introduced a text mining based indexing method for symbolic representation of musical data that extracts maximal frequent phrases from musical data and sorts them by their length, frequency and personality.

Finally, the paper by Grachten et al. (2004) is of particular relevance because it represents melodies at a higher level than notes but lower enough to capture the essence of the melody. This level is the 'Narmour patterns' level, based on Narmour's I/R model (1992), which is well-known in musicology.

Our algorithm is also based in a symbolic representation of the song: MusicXML.

2.2 Sequence mining

In our approach for musical motif extraction, we first transform a song into a sequence of notes. There is a rich variety of sequence types, ranging from simple sequences of letters to complex sequences of relations.

A *sequence* over an element type τ is an ordered list $S = s_1 ... s_m$, where:

- each s_i (which can also be written as $S[i]$) is a member of τ, and is called an element of S;
- m is referred to as the length of S and is denoted by $|S|$;
- each number between 1 and $|S|$ is a position in S.

$T = t_1 .... t_n$ is called a *subsequence* of the sequence $S = s_1 ... s_m$ if there exist integers $1 < j_1 < j_2 < ... < j_n < m$ such that $t_1 = s_{j_1}$, $t_2 = s_{j_2}$, and in general, $t_n = s_{j_n}$.

Sequences have been used to solve different problems in the literature (Dong and Pei 2007; Han and Kamber 2005):

- *String matching problems*: Several sequence mining techniques have been used, for instance, in Bioinformatics to find some structures in a DNA sequence (Böckenhauer and Bongartz 2007):
  - Exact string matching: Given two strings, finding the occurrence of one as a substring of the other one.
  - Substring search: Finding all the sequences in a sequence database that contain a particular string as a subsequence.
  - Longest common substring: Finding the substring with maximum length that is common to all the sequences in a given set.
  - String repetition: Finding substrings that appear at least twice in a sequence.
- *Periodic pattern discovery*: A traditional periodic pattern consists of a tuple of k components, each of which is either a literal or '*', where k is the period of the pattern and '*' can be substituted for any literal and is used to enable the representation of partial periodicity (Wang et al. 2001; Yang et al. 2001).
- *Sequence motifs:* A motif is essentially a short distinctive sequential pattern shared by a number of related sequences. There are four main problems in this area (Dong and Pei 2007): motif representation (i.e., designing the proper motif representation for the different applications), motif finding (i.e., finding the motifs shared by several sequences), sequence scoring (i.e., computing the probability of a sequence to be generated by a motif—using Markov models, for example), and sequence explanation (i.e., given a sequence and a motif with hidden states, providing the most likely state path that produced that sequence).
- *Sequential pattern mining in transactional databases:* Sequential patterns have been used for predicting the behavior of individual customers. Each customer is typically modeled by a sequence of transactions containing the set of items he has bought. Several algorithms address this kind of problems, the most common being AprioriAll (Agrawal and Srikant 1994), SPADE (Zaki 2001) GSP (Srikant and Agrawal 1996), and PrefixSpan (Pei et al 2001).

- *Sequential pattern mining in sequence databases*: Some algorithms, such as the one developed by Jiang and Hamilton (2003), look for subsequences that have a larger frequency (or number of repetitions) than an user-defined threshold, which is established beforehand. Jiang's algorithm, for example, uses a tree-based structure called *trie*, that preserves the number of times a subsequence is present in the sequence. Three different versions of his algorithm can be devised:
  - A breadth-first search algorithm passes K times through the data sequence, counting the sequences of size i in its i-th iteration. At the end of each iteration, infrequent subsequences are pruned.
  - A depth-first search algorithm passes just one time through the data using a window of size K which is moved one position at a time. The sequence in the window is preserved in the *trie* structure as well as its prefix. The way this *trie* structure is completely built is very memory consuming without pruning.
  - A heuristic-first search algorithm is a variation of the depth-first algorithm. The number of occurrences of the prefixes of a subsequence is compared to the threshold before inserting the subsequence in the *trie*. If any prefix of the subsequence has not yet been shown to be frequent, then occurrences of the subsequence itself are not counted. This algorithm is more efficient in time and space but it is not able to find all the frequent subsequences.

Our problem of motif extraction in a piece of music could be seen as a particular case of the 'sequential pattern mining in sequence databases' problem. However, the algorithms proposed by Jiang et al. have the drawback that they are limited by the size of the alphabet (i.e., the element type τ according to our notation). This value can be very high in our particular domain, provided that we take different note pitches and durations into account.

Furthermore, we consider the presence of similar sequences (transposed motifs in our problem) that should be counted as if they were exact repetitions. In our case, it is also interesting to know where these repetitions appear in the sequence, specially when considering these similar repetitions. This information would not be given by Jiang's algorithms, as they only count the number of repetitions.

In the following section, we propose a novel TreeMiner-based algorithm to find motifs in a sequence in order to solve the problem of motif extraction in a piece of music. It should be noted, however, that our algorithm can be applied to several sequences at a time, as well as to different kinds of sequence databases (not just musical ones).

3 Our sequence pattern mining algorithm

The goal of frequent sequence pattern mining is the discovery of all the frequent subsequences in a large database of sequences D or in an unique large sequence.

Let $\delta_T(S)$ be the occurrence count of a subsequence S in a sequence T and d_T a variable such that $d_T(S) = 0$ if $\delta_T(S) = 0$ and $d_T(S) = 1$ if $\delta_T(S) > 0$. We define the *support* of a subsequence as $\sigma(S) = \sum_{T \in D} d_T(S)$, i.e., the number of sequences in D that include at least one occurrence of the subsequence S. Analogously, the *weighted support* of a subsequence is defined as $\sigma_w(S) = \sum_{T \in D} \delta_T(S)$, i.e., the total number of occurrences of S within all the sequences in D.

We also consider the occurrences of a pattern that approximately match (i.e., those occurrences that are very similar but are not exactly the same). We define the *exact support* of a subsequence as the number of occurrences that are exactly equal to the pattern, whereas the *transposed support* includes both exact and similar occurrences.

We say that a subsequence S is *frequent* if its support is greater than or equal to a predefined minimum support threshold. We define L_k as the set of all frequent k-subsequences (i.e., subsequences of size k).

3.1 SSMiner

Our algorithm, called SSMiner (Similar Sequence Miner), is based on the POTMiner (Jimenez et al. 2009) frequent tree pattern mining algorithm, a TreeMiner-like algorithm for discovering frequent patterns in trees (Zaki 2005b). POTMiner and its antecessor follow the Apriori (Agrawal and Srikant 1994) iterative pattern mining strategy, where each iteration is broken up into two distinct phases:

– *Candidate Generation*: A candidate is a potentially frequent subsequence. In Apriori-like algorithms, candidates are generated from the frequent patterns discovered in the previous iteration. Most Apriori-like algorithms, including ours, generate candidates of size $k+1$ by merging two patterns of size k having $k-1$ elements in common.
– *Support Counting*: Given the set of potentially frequent candidates, this phase consists of determining their actual support and keeping only those candidates whose support is above the predefined minimum support threshold (i.e., those candidates that are actually frequent).

The pseudo-code of our algorithm is shown in Fig. 1 and its implementation details will be discussed in Sections 3.2 through 3.4.

The sequence of a song is scanned twice by our algorithm, in the process of obtaining the frequent elements of size 1. The first scan is needed to save the occurrences of each note and the second one is employed to detect the transposed occurrences of each note. Then, the infrequent notes are pruned and we are ready to apply the two phases of the SSMiner algorithm without checking the original sequence any more.

Fig. 1 SSMiner: our sequence mining algorithm

```
algorithm
   Obtain frequent elements (frequent patterns of size 1)
   Build candidate classes C_1 from the frequent elements
   for k=2 to MaxSize
      for each class P ∈ C_{k-1}
         for each element p ∈ P.
            Compute the frequency of p
            if p is frequent
            then
               Create a new class P′ from p.
               Add P′ to C_k
```

3.2 Candidate generation

We use an equivalence class-based extension method to generate candidates (Zaki 2005a). This method generates $(k+1)$-subsequence candidates by joining two frequent k-subsequences with $k-1$ elements in common.

Two k-subsequences are in the same equivalence class $[P]$ if they share the same prefix string until their $(k-1)$th element. Each element of the class can then be represented as x, where x is the k-th element label.

Elements in the same equivalence class are joined to generate new candidates. This join procedure, called extension in the literature, works as follows. Let (x) and (y) denote two elements in the same class $[P]$, and $[P_x]$ be the set of candidate sequences derived from the sequence that is obtained by adding the element (x) to P. The join procedure results in attaching the element (y) to the sequence generated by adding the element (x) to P, i.e, $(y) \in [P_x]$. Likewise, $(x) \in [P_y]$

3.3 Occurrence lists

Once we have generated the potentially frequent candidates, it is necessary to determine which ones are actually frequent.

The support counting phase in our algorithm follows the strategy of AprioriTID (Agrawal and Srikant 1994). Instead of checking the presence of each candidate in the sequence (which would entail $O(|S|)$ operations), special lists are used to preserve the occurrences of each pattern in the database, thus facilitating the support counting phase.

Each occurrence list contains tuples (t, m, p, d, Θ) where t is the sequence identifier, m stores the elements of the sequence which match those of the $(k-1)$ prefix of the pattern X, p is the position of the last element in the pattern X, d is a position-based parameter used for guaranteeing that elements in the pattern are contiguous within the sequence and Θ indicates the similarity between the occurrence and the original pattern.

When building the scope lists for patterns of size 1, m is empty and the element d is initialized with the position of the pattern only element in the original database sequence. In the first pass through the sequence, exact patterns of size 1 are collected, being its Θ parameter initialized as "=". When similar pattern occurrences are collected in the second pass through the sequence, the parameter Θ is initialized with a value that indicates the similarity between the original pattern and the actual occurrence.

We obtain the occurrence list for a new candidate of size k by joining the lists of the two subsequences of size $k-1$ that were involved in the generation of the candidate. Let $(t_x, m_x, p_x, d_x, \Theta_x)$ and $(t_y, m_y, p_y, d_y, \Theta_y)$ be two tuples to be joined. The join operation proceeds as follows:

if

1. $t_x = t_y = t$ **and**
2. $m_x = m_y = m$ **and**
3. $d_x = 1$ (only if $k \neq 2$) **and**
4. $p_x < p_y$ **and**
5. Θ_x=Θ_y

then add $[t,\ m \bigcup \{p_x\},\ p_y,\ d_y - d_x,\ \Theta_y]$ to the occurrence list of the generated candidate.

3.4 Support counting

Checking if a pattern is frequent consists of counting the elements in its occurrence list. The counting procedure is different depending on whether the weighted support σ_w is considered or not.

- If we count occurrences using the weighted support, all the tuples in the lists must be taken into account.
- If we are not using the weighted support, the support of a pattern is the number of different sequence identifiers within the tuples in the occurrence list of the pattern.

It should be noted that d represents the distance between the last node in the pattern and its prefix m. Therefore, we only have to consider the elements in the scope lists whose d parameter equals 1 for guaranteeing that elements in the pattern are contiguous within the sequence. It is important to remark that the remaining elements in the lists cannot be eliminated because they are needed to build the occurrence lists of larger patterns.

3.5 Representative patterns

Our algorithm returns all the frequent patterns of the maximum size indicated by the user (or smaller ones if there are no patterns of such size). As musical motifs are generally no longer than a measure, a value of ten is typically used by default. Nevertheless, this limit can be easily modified since our algorithm can return all the frequent patterns that exist in the song regardless of their size. The resulting output will be the set of frequent patterns that represent the song. The algorithm also returns the positions of the different occurrences of the patterns within the song (including transposed occurrences if needed).

3.6 SSMiner complexity

SSMiner starts by computing the frequent patterns of size 1. This step is performed by obtaining the vertical representation of the sequence database, i.e., the individual notes that appear in the sequences with their occurrences represented as scope lists. This representation is obtained in linear time with respect to the number of sequences in the database just by scanning it and building the scope lists for patterns of size 1. We then discard the patterns of size 1 that are not frequent. This results in L scope lists corresponding to the L frequent notes in the sequence database and each frequent label leads to a candidate class of size 1.

Let $c(k)$ be the number of classes of size k, which equals the number of frequent patterns of size k, and $e(k)$ the number of elements that might belong to a particular class of size k (i.e., the number of patterns of size $k+1$ that might be included in the class corresponding to a given pattern of size k).

In SSMiner, each sequence pattern grows only by adding an element at the end of the sequence pattern. The number of different sequences of size $k+1$ that can be obtained by the extension of a sequence of size k is $L \cdot k$. Hence, the number of elements in a particular class, $e(k)$, is $O(L)$.

The number of classes of size 1 equals L, the number of frequent labels, so that $c(1) = L$. The classes of size $k+1$ are derived from the frequent elements in classes of size k. In the worst case, when all the $e(k)$ elements are frequent, $c(k+1) = c(k) \cdot e(k)$. Solving the recurrence, we obtain $c(k+1) = c(k) \cdot L = O(L^k)$.

For each considered pattern of size $k+1$, SSMiner must perform a join operation to obtain its scope list from the scope lists of the two patterns of size k that led to it.

The size of the scope list for a pattern of size k is $O(t \cdot e)$ while the computational cost of a scope-list join operation is $O(t \cdot e^2)$, where t is the number of sequences in the database and e is the average number of embeddings of the pattern in each sequence (Zaki 2005a).

In the worst case, the number of embeddings $s(k-1)$ of a pattern of size $k-1$ in a sequence of size S equals the number of subsequences of size $k-1$ within the sequence of size S. This number is bounded by $S-k+1$.

Hence, the cost of the join operation needed for obtaining the scope list of a pattern of size k, is $j(k) = O(t \cdot s(k-1)^2) = O(t \cdot (S-k+1)^2)$.

The cost of obtaining all the frequent patterns of size k will be, therefore, $O(c(k) \cdot j(k)) = O(L^k \cdot t \cdot (S-k+1)^2)$

The total cost of executing the SSMiner algorithm to obtain all the frequent patterns up to $k = MaxSize$ is $\sum_{k=1 \ldots MaxSize}(L^k \cdot t \cdot (S-k+1)^2)$. Since the running time of our algorithm is dominated by the time needed to discover the largest patterns (i.e., $k = MaxSize$), SSMiner is $O(L^{MaxSize} \cdot t \cdot (S - MaxSize + 1)^2)$.

Therefore, our algorithm execution time is proportional to the number of sequences in the sequence database ($t = 1$ in our motifs identification problem), and to the number of patterns than can be identified (L^k). Finally, its execution time is quadratic with respect to the size of the sequences (S).

4 An example

In this section, we present an example to help the reader understand the way our algorithm identifies frequent subsequences in a sequence. In order to facilitate the understanding of the procedure, we are not considering the duration of notes. Furthermore, we only take into account those transpositions of fifth.

We will use in this paper the scientific pitch notation which combines a letter-name, accidentals (if any) and a number identifying the pitch's octave. This notation is the most common in English written texts.

Let's suppose we have the following piece of a song: G4 A4 G4 E4 D5 E5 D5 B4 G4 A4 G4 E4 A4 G4 B4 G4 A4 G4 E4 (see Fig. 2), and we want to extract those subsequences that appear at least four times in it.

The first step of our algorithm is scanning the sequence to obtain all the occurrences of each note. Then, the occurrence lists of each note are built as indicated in Section 3.3. Results are shown in Fig. 3.

Fig. 2 Sample piece: G4 A4 G4 E4 D5 E5 D5 B4 G4 A4 G4 E4 A4 G4 B4 G4 A4 G4 E4

The first element is 1 in all the tuples because we only have one sequence (i.e., only one song) in our example. The second one is the prefix of the substring (empty in patterns of size 1). The third element indicates the position of the last element of the pattern in the sequence. The fourth element is the distance between the last element of the pattern and its prefix (or the position of the element when there is no prefix, as this is the case). Finally, the last element indicates if the occurrence is exactly equal to the pattern ('=') or if it is transposed.

In this example, only those transpositions of up one fifth are being taken into account, so that '+5' is the only alternative for this element in our example. It should be noted, however, that all possible transpositions—distances between two notes—could be taken into account. In any case, we need only to compare notes in one direction, as we will always find at least a version of the pattern that summarizes all its transpositions.

Going back to our example, the note G4 is transposed up one fifth as D5. Therefore, there are 9 tuples in the occurrence list of G4: 7 as itself, 2 as D5.

The next step is checking if all the notes are frequent. In this case, only G4, A4, and E4 have at least four occurrences. Therefore, only these patterns will be kept.

Figure 4 shows the extension of the element G4. This element is extended with all the frequent patterns of size 1 including itself, and the occurrence lists of each candidate pattern of size 2 are obtained by joining the lists of the elements that generated it, as explained in Section 3.3.

Figure 4 shows, with bold letters, the tuples where $d = 1$. That means that these are contiguous occurrences of the pattern. In our example, only the patterns G4 A4 and G4 E4 appear as contiguous subsequences in our song. Furthermore, they have at least four occurrences—our minimum support threshold—and they will be extended to generate candidates of size 3. It should be noted that the pattern G4 G4

G4	A4	E4	D5	E5	B4
{1,_,1,1,=}	{1,_,2,2,=}	{1,_,4,4,=}	{1,_,5,5,=}	{1,_,6,6,=}	{1,_,8,8,=}
{1,_,3,3,=}	{1,_,10,10,=}	{1,_,12,12,=}	{1,_,7,7,=}		{1,_,15,15,=}
{1,_,9,9,=}	{1,_,13,13,=}	{1,_,19,19,=}			
{1,_,11,11,=}	{1,_,17,17,=}	{1,_,8,8,+5}			
{1,_,14,14,=}	{1,_,6,6,+5}	{1,_,15,15,+5}			
{1,_,16,16,=}					
{1,_,18,18,=}					
{1,_,5,5,+5}					
{1,_,7,7,+5}					

Fig. 3 Occurrence lists of the elements of the following sequence: G4 A4 G4 E4 D5 E5 D5 B4 G4 A4 G4 E4 A4 G4 B4 G4 A4 G4 E4

Prefix: G4

G4 G4		G4 A4		G4 E4	
{1,1,3,2,=}	{1,1,9,8,=}	**{1,1,2,1,=}**	{1,1,10,9,=}	{1,1,4,3,=}	{1,1,12,11,=}
{1,1,11,10,=}	{1,1,14,13,=}	{1,1,13,12,=}	{1,1,17,16,=}	{1,1,19,18,=}	**{1,3,4,1,=}**
{1,1,16,15,=}	{1,1,18,17,=}	{1,3,10,7,=}	{1,3,13,10,=}	{1,3,12,9,=}	{1,3,19,16,=}
{1,3,9,6,=}	{1,3,11,8,=}	{1,3,17,14,=}	**{1,9,10,1,=}**	{1,9,12,3,=}	{1,9,19,10,=}
{1,3,14,11,=}	{1,3,16,13,=}	{1,9,13,4,=}	{1,9,17,8,=}	**{1,11,12,1,=}**	{1,11,19,8,=}
{1,3,18,15,=}	{1,9,11,2,=}	{1,11,13,2,=}	{1,11,17,6,=}	{1,14,19,5,=}	{1,16,19,3,=}
{1,9,14,5,=}	{1,9,16,7,=}	{1,14,17,3,=}	**{1,16,17,1,=}**	**{1,18,19,1,=}**	{1,5,8,3,+5}
{1,9,18,9,=}	{1,11,14,3,=}	**{1,5,6,1,+5}**		{1,5,15,10,+5}	**{1,7,8,1,+5}**
{1,11,16,5,=}	{1,11,18,7=}			{1,7,15,8,+5}	
{1,14,16,2,=}	{1,14,18,4,=}				
{1,16,18,2,=}	{1,5,7,2,+5}				

Fig. 4 Extension of the element G4 in Fig. 3

is not contiguous and will not be extended. However, it is preserved to perform the extension of G4 A4 with G4 E4.

After two more extensions, which are done in the same way, we obtain the pattern G4 A4 G4 E4 with a *support* of 4 and an *exact support* of 3. This is be the pattern we would use to characterize our example song.

5 Experiments

We have tested our algorithm using a corpus of 44 songs. This set includes songs from a wide variety of authors. The first column in Table 1 shows the songs used in our experiments.

We have performed 4 experiments with different constraints:

- Exact pitch and duration (**pitch-duration**)
- Exact pitch and any rhythm (**pitch**)
- Transpositions but exact duration (**transposition-duration**)
- Transpositions and any duration (**transposition**)

Pitch-duration is the most restrictive one, whereas **transposition** is the experiment with a lower number of constraints. All these configurations are representative when looking for musical motifs, as they can be modified in tempo or in pitch. Unlike the example in the former section, all the possible transpositions are taken into account in these experiments.

Table 1 summarizes the results of our experiments. Each row corresponds to one of the songs in the corpus. The second column (*'notes'*) indicates the number of notes in each song.

The *'Max Size'* column indicates the size of the longest pattern(s) found in each song. Patterns of this size are the only ones that are finally returned to the user. As our aim in this paper is searching for repeating motifs, and not whole repeating sections, we have to introduce an upper limit to the size of the patterns. Preliminary

Table 1 Corpus of songs and results for each song in our four series of experiments

Song	Notes	Pich-duration			Pich			Transposition-duration				Transposition			
		Max S	# pat	r_sup	Max S	# pat	r_sup	Max S	# pat	r_sup	t_sup	Max S	# pat	r_sup	t_sup
Alfie	192	7	1 / 41	4	7	1 / 45	5	7	1 / 120	4	4	7	1 / 135	5	5
All good things	248	9	2 / 97	[4–5]	9	2 / 103	[4–5]	9	2 / 151	[4–5]	[4–5]	9	2 / 183	[4–5]	[4–5]
Angels	388	8	1 / 98	4	10	3 / 195	4	8	1 / 232	4	4	10	3 / 267	4	4
Angie	530	6	2 / 106	[4–5]	10	4 / 188	4	6	2 / 401	[4–5]	[4–5]	10	4 / 455	4	4
Apologize	384	10	9 / 217	4	10	10 /209	4	10	9 / 318	4	4	10	10 / 283	4	4
Bach	182	5	2 / 38	4	5	2 / 44	4	10	2 / 178	2	5	9	1 / 189	2	4
Ballade pour Adeline	453	10	33 / 376	[4–12]	10	36 / 353	[4–30]	10	33 / 566	[4–18]	[4–27]	10	36 / 528	[4–36]	[4–52]
Beat it	171	4	5 / 42	[4–5]	6	1 / 51	4	4	5 / 62	[4–5]	[4–6]	7	1 / 68	2	4
Beautiful	332	9	1 / 82	4	10	1 / 112	4	9	1 / 331	4	4	10	1 / 265	4	4
Beautiful that way	268	10	20 / 243	4	10	20 / 235	[4–5]	10	20 / 316	4	4	10	20 / 356	4	4
Bleeding love	539	9	1 / 148	4	9	2 / 185	[4-5]	9	1 / 277	4	4	9	2 / 320	[4–5]	[4–5]
Brown eyed girl	138	3	3 / 25	[4–7]	5	1 / 34	4	4	1 / 54	3	4	5	2 / 71	[3–4]	[4–5]
Crazy	307	5	2 / 81	4	6	2 / 85	4	6	2 / 285	[1–3]	4	6	3 / 201	[3–4]	4
Don't speak	334	9	1 / 97	4	10	2 / 145	4	9	1 / 244	4	4	10	2 / 217	4	4
Every breath you take	295	6	1 / 56	7	6	1 / 68	4	8	1 / 241	3	5	8	4 / 265	[3–4]	[4–5]
Everybody hurts	271	4	1 / 46	4	8	3 / 83	4	4	2 / 121	[1–4]	4	6	1 / 149	4	4
Feel	394	10	9 / 159	[5–6]	10	13 / 172	4	10	9 / 393	[5–6]	[5–6]	10	13 / 364	[4–7]	[4–7]
Fever	121	8	1 / 31	4	8	1 / 37	[4–7]	8	1 / 45	4	4	8	1 / 54	4	4
Fur Elise	267	10	16 / 202	[4–6]	6	16 / 77	4	10	16 / 396	[4–6]	[4–6]	10	16 / 391	[4–6]	[4–6]
Hero	259	5	1 / 56	4	10	1 / 141	5	6	1 / 220	1	4	6	1 / 214	5	5
How to save a life	448	7	1 / 102	5	10	3 / 100	5	7	1 / 251	5	5	10	1 / 226	5	5
I hate this part	231	10	3 / 94	4	10	6 / 103	[4–7]	10	6 / 301	[1–4]	[4–5]	10	6 / 333	[1–4]	[4–5]
I say a little prayer	172	4	2 / 32	4	4	3 / 39	4	5	2 / 95	3	[4–5]	5	1 / 92	2	5
Imagine	188	5	1 / 44	5	6	1 / 49	[4–5]	6	1 / 93	1	4	10	6 / 150	[4–5]	[4–5]

Let it be	179	5	1 / 32	4	10	1 /49	4	5	1 / 84	4	4	6	1 / 129	4	4
Livin la Vida Loca	434	10	19 / 342	[4–6]	5	22 /345	[4–8]	10	19 / 507	[4–6]	[4–6]	10	23 / 455	[2–8]	[4–8]
Love me tender	53	2	2 / 7	[4–6]	4	1 / 33	4	5	1 / 28	1	4	5	2 / 33	[1–4]	4
Paint it black	129	2	3 / 19	[4–6]	10	16 / 205	5	3	4 / 51	2	[4–6]	5	1 / 48	2	4
Quizas quizas quizas	156	8	1 / 61	6	9	1 / 70	[4–6]	8	1 / 146	6	6	9	1 / 135	6	6
Roxanne	396	10	18 / 213	[4–10]	10	20 / 226	6	10	18 / 270	[4–10]	[4–10]	10	20 / 256	[4–10]	[4–10]
Smile	105	4	2 / 30	[4–6]	4	4 / 33	[4–10]	10	4 / 211	2	4	10	1 / 91	2	4
Sunshine of my life	352	10	26 / 308	[5–6]	10	27 / 305	[5–6]	10	26 / 419	[5–6]	[5–6]	10	27 / 409	[5–6]	[5–6]
The nearness of you	125	4	1 / 24	4	5	3 / 36	4	5	1 / 82	1	4	5	4 / 90	[1–4]	4
Torn	388	10	11 / 194	[4–5]	10	11 / 200	[4–5]	10	11 / 275	[4–5]	[4–5]	10	11 / 271	[4–5]	[4–5]
Under the bridge	541	10	9 / 246	4	10	14 / 291	4	10	9 / 661	4	4	10	14 / 889	[4–6]	[4–6]
Underneath your clothes	260	5	1 / 55	4	5	3 / 67	[4–6]	5	1 / 206	4	4	7	1 / 224	2	4
Viva la vida	444	10	1 / 165	4	10	13 / 235	[4–5]	10	1 / 366	4	4	10	13 / 476	[4–5]	[4–5]
We're the champions	303	5	1 / 62	4	8	1 / 93	4	5	1 / 206	4	4	8	2 / 256	[2–4]	4
What a wonderful world	179	7	1 / 54	4	8	1 / 61	6	7	1 / 149	4	4	8	1 / 109	6	6
When I'm sixtyfour	185	6	1 / 30	4	7	1 / 41	4	7	1 / 116	1	4	7	2 / 186	[1–4]	[4–5]
With or without you	236	5	1 / 44	4	5	1 / 57	4	5	1 / 204	4	4	5	2 / 165	[1–4]	[4–5]
Wonderwall	184	9	1 / 68	4	9	1 / 75	4	9	1 / 121	4	4	9	1 / 135	4	4
Your song	236	5	1 / 43	4	5	1 / 51	4	5	1 / 118	4	4	5	1 / 96	4	4
Zombie	239	10	3 / 92	4	10	3 /100	4	10	3 / 130	4	4	10	3 / 141	4	4

tests with our song corpus have shown that a value of ten is adequate for this parameter providing that musical motifs are generally no longer than a measure.

The *'Patterns'* column indicates the number of patterns our algorithm finds. The first number is the amount of patterns of maximum size (i.e. the size indicated on the previous column); whereas the second number is the total amount of patterns, regardless of their size. As the reader can see, the total amount of patterns is pretty high. However, many of these patterns are not musically relevant since they are part of bigger ones. Even more, patterns of size 1 and 2, which can hardly be considered as motifs, are also included in this set.

The *'exact support'* column is the exact support of the returned patterns. Intervals appear in some cases due to the fact that not all the returned patterns necessarily appear the same number of times.

Finally, *'transposed support'* indicates the support of patterns including transpositions. This column is only relevant for the experiments **transposition-duration** and **transposition**, when transpositions are taken into account. As expected, values in this column are always equal or greater than those in the corresponding *'exact support'* column.

For us, the minimum support required for a pattern to be considered frequent is four. Any pattern with a lower number of repetitions will be deleted. This value has been manually set regarding the size of the evaluating songs and some preliminary tests. However, it can be easily adjusted when new datasets require it.

The reader can also notice that, in some cases (namely, when transpositions are taken into account), patterns with *exact support* lower than four can be found. Those patterns do not have enough exact repetitions as themselves, but they are frequent when transpositions are taken into account. Hence, transpositions are important because, without considering them, some patterns would have not been discovered, as they do not reach the minimum support threshold just by exact repetitions. That happens, for instance, with the 'Crazy' and 'Hero' songs.

Regarding the length of the excerpts that have been tested, three notes can hardly represent any meaningful motif. However, almost anyone could identify Beethoven's Fifth Symphony by just four notes. Hence, a minimum length of four seems adequate.

It should be noted that, in some situations, there is another transposition of the pattern that has greater *exact support* than the one returned by our algorithm as described in Section 3. As we mentioned earlier, our algorithm looks for transposed motives only in one direction and this suffices to guarantee that it will find all the relevant occurrences; however, the returned patterns are not necessarily the most frequent exact ones. Given that our algorithm keeps track of the occurrences of a given pattern and all of its transpositions, it is trivial to obtain the most frequent exact pattern just by looking at the corresponding scope list. This pattern will correspond to the most common Θ in the scope list.

Table 2 Percentage of songs that include at least one identified pattern within their chorus

	Pitch-duration	Pitch	Transposition-duration	Transposition
%Yes	63.64	68.18	63.64	72.73
%No	29.55	25.00	29.55	20.45
%Without chorus	6.82	6.82	6.82	6.82

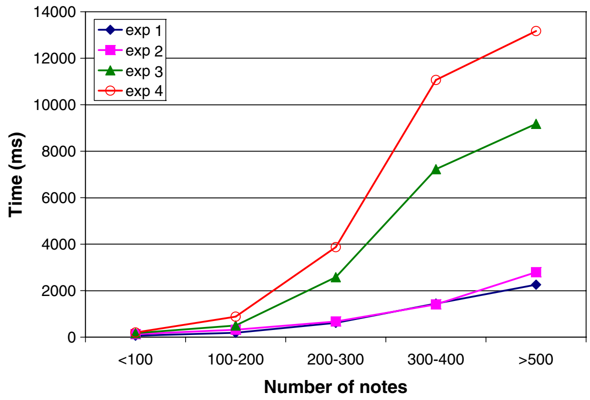


Fig. 5 SSMiner execution time

It should also be observed that, in some songs, the number of patterns of *MaxSize* elements is pretty high (e.g. 'Ballade pour Adeline' or Beethoven's 'Für Elise'). This is due to the fact that many of those are still subpatterns of bigger ones. For instance, a pattern of size 15 includes six sequential subpatterns of size 10 (starting from the 1st, 2nd, 3rd, 4th, 5th and 6th note, respectively).

In order to evaluate the goodness of our method, we have checked whether or not the discovered frequent patterns belong to the chorus of the song—in our experiments, 6.82% of the songs do not have a clear chorus. Table 2 shows the percentage of songs which have at least one identified pattern within their chorus. As can be seen, above 60% of the songs fulfill this requirement. Also, it is remarkable that not considering the rhythm results in more patterns belonging to the chorus of the songs. This fact indicates that patterns are not always exactly repeated as themselves, but slightly modified. Although the chorus-belonging criterion appears to be a valid and obvious one, it should be noted that some songs are better identified by patterns which do not belong to the chorus.

Regarding SSMiner computation time, Fig. 5 shows the time consumed in each experiment with respect to the number of notes in the melody. The chart groups the songs into five groups according to their lengths and displays the average execution time for each subset of melodies. These execution times are quadratic with respect to the number of notes in the melodies, as explained in Section 3.6.

6 Conclusions

We have presented the application of frequent pattern mining to the discovery of musical motifs in a piece of music. *MusicXML* files, which can be easily collected, are transformed into sequences of notes, defined at their lower level. Our algorithm, SSMiner, is able to efficiently identify frequent subsequences in a sequence.

The matching between the patterns does not need to be exact. Our algorithm is able to identify transposed patterns including exact matchings, i.e., null transpositions. Our experiments suggest that our approach performs well in a set of randomly-selected songs.

In the future, we intend to employ interval strings to represent notes rather than the absolute pitches we have used in the experiments reported in this paper. We will also consider more abstract representations of melodies, such as the one proposed by Narmour. Finally, we plan to study the parallelization of our algorithm implementation in order to improve its execution time, which is already asymptotically optimal.

Acknowledgements F. Berzal and A. Jiménez are supported by the projects TIN2006-07262 and TIN2009-08296, whereas W. Fajardo and M. Molina-Solana are supported by the research project TIN2006-15041-C04-01.

References

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *20th int. conf. on very large data bases* (pp. 487–499).

Aucouturier, J. J., & Sandler, M. (2002). Finding repeating patterns in acoustic musical signals: Applications for audio thumbnailing. In *Audio engineering 22nd int. conf. on virtual, synthetic and entertainment audio (AES22)* (pp. 412–421).

Bartsch, M., & Wakefield, G. (2005). Audio thumbnailing of popular music using chroma-based representations. *IEEE Transactions on Multimedia, 7*(1), 96–104.

Berzal, F., Fajardo, W., Jiménez, A., & Molina-Solana, M. (2009). Mining musical patterns: Identification of transposed motives. In *18th Int. symposium of foundations of intelligent systems*. Lecture Notes in Computer Science, vol. 5722, pp. 271–280.

Böckenhauer, H. J., & Bongartz, D. (2007). *Algorithmic aspects of bioinformatics*. New York: Springer.

Cambouropoulos, E., Crawford, T., & Iliopoulos, C. S. (2001). Pattern processing in melodic sequences: Challenges, caveats and prospects. *Computers and the Humanities, 35*(1), 9–21.

Chu, S., & Logan, B. (2002). Music summary using key phrases. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP-00)* (pp. 749–752).

Dong, G., & Pei, J. (2007). *Sequence data mining (advances in database systems)*. New York: Springer.

Grachten, M., Arcos, J. L., & de Mantaras, R. L. (2004). Melodic similarity: Looking for a good abstraction level. In *5th Int. Conf. on Music Information Retrieval (ISMIR 2004)* (pp. 210–215).

Han, J., & Kamber, M. (2005). *Data mining: Concepts and techniques*. Denver: Morgan Kaufmann.

Hsu, J. L., Liu, C. C., & Chen, A. (1998). Efficient repeating pattern finding in music databases. In *ACM 7th int. conf. on information and knowledge management* (pp. 281–288).

Jiang, L., & Hamilton, H. J. (2003). Methods for mining frequent sequential patterns. In *Advances in artificial intelligence, Lecture of Notes in Computer Sciences* (Vol. 2671/2003, pp. 486–491). Berlin: Springer.

Jimenez, A., Berzal, F., & Cubero, J. C. (2009). Mining induced and embedded subtrees in ordered, unordered, and partially-ordered trees. *Knowledge and Information Systems, 4994/2008*, 111–120. doi:10.1007/s10115-009-0213-3.

Levy, M., & Sandler, M. (2008). Structural segmentation of musical audio by constrained clustering. *IEEE Transactions on Audio, Speech, and Language Processing, 16*(2), 318–326.

Meredith, D., Lemström, K., & Wiggins, G. A. (2002). Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research, 31*(4), 321–345

Narmour, E. (1992). *The analysis and cognition of melodic complexity: The implication realization model*. Chicago: Univ. Chicago Press.

Paulus, J., & Klapuri, A. (2009). Music structure analysis using a probabilistic fitness measure and a greedy search algorithm. *IEEE Transactions on Audio, Speech, and Language Processing, 17*(6), 1159–1170.

Pei, J., Han, J., Asl, M. B., Pinto, H., Chen, Q., Dayal, U., et al. (2001). Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *5th int. conf. on extending database technology* (pp. 215–224).

Pienimäki, A. (2002). Indexing music databases using automatic extraction of frequent phrases. In *3rd int. conf. on music information retrieval* (pp. 25–30).

Rolland, P. Y. (1998). Discovering patterns in musical sequences. *Journal of New Music Research, 28*(4), 334–350

Srikant, R., & Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements. *Extending Database Technology, 1057*, 3–17.
Wang, W., Yang, J., & Yu, P. S. (2001). Meta-patterns: Revealing hidden periodic patterns. In *IBM research report* (pp. 550–557).
Yang, J., Wang, W., & Yu, P. S. (2001). Infominer: mining surprising periodic patterns. In *7th ACM int. conf. on knowledge discovery and data mining (SIGKDD)* (pp. 395–400). New York: ACM
Zaki, M. J. (2001). Spade: an efficient algorithm for mining frequent sequences. *Machine Learning, 42*, 31–60.
Zaki, M. J. (2005a) Efficiently mining frequent embedded unordered trees. *Fundamenta Informaticae, 66*(1–2), 33–52
Zaki, M. J. (2005b). Efficiently mining frequent trees in a forest: Algorithms and applications. *IEEE Transactions on Knowledge and Data Engineering, 17*(8), 1021–1035.
Zhang, T., & Samadani, R. (2007). Automatic generation of music thumbnails. In *Proceedings of the 2007 IEEE int. conf. on multimedia and expo* (pp. 228–231).

Representation model and learning algorithm for uncertain and imprecise multivariate behaviors, based on correlated trends

Delgado, M., Fajardo, W. & Molina-Solana, M. (submitted), "*Representation model and learning algorithm for uncertain and imprecise multivariate behaviors, based on correlated trends*", Information Sciences.

- Status: **Submitted**
- Impact Factor (JCR 2010): 2.836
- Subject category:
  - COMPUTER SCIENCE, INFORMATION SYSTEMS (10/128 Q1)

Elsevier Editorial System(tm) for Information Sciences
Manuscript Draft

Manuscript Number: INS-D-12-233

Title: Representation model and learning algorithm for uncertain and imprecise multivariate behaviors, based on correlated trends

Article Type: Full length article

Keywords: behaviour modelling; dataseries; Frequent Correlated Trends

Corresponding Author: Mr. Miguel Molina-Solana, MSc

Corresponding Author's Institution: Universidad de Granada

First Author: Miguel Delgado, Professor

Order of Authors: Miguel Delgado, Professor; Waldo Fajardo; Miguel Molina-Solana, MSc

Suggested Reviewers: José Manuel Molina López PhD in Telecommunications
Full Professor, Computer Science and Engineering Department, Universidad Carlos III de Madrid
molina@ia.uc3m.es

Hani Hagras BSc, MSc, PhD, SMIEEE, FIET
Director of the Computational Intelligence Centre, University of Essex, UK
hani@essex.ac.uk

James M. Keller PhD in Mathematics
Professor, Computer Engineering and Computer Science Department, University of Missouri-Columbia
kellerj@missouri.edu

Representation model and learning algorithm for uncertain and imprecise multivariate behaviors, based on correlated trends

Miguel Delgado, Waldo Fajardo, Miguel Molina-Solana*

Department of Computer Science and AI. Universidad de Granada

Abstract

The computational representation and classification of behaviors is a task of growing interest in the field of *Behavior Informatics*, being series of data a common way of describing them. However, as these data are often imperfect, new representation models are required in order to handle that imperfection. This work presents a new approach, *Frequent Correlated Trends*, for representing uncertain and imprecise multivariate data series. Such a model can be applied to any domain where behaviors recur in similar —not identical— shape, and we have already used them to the task of identifying the performers of violin recordings with good results.

Keywords: behaviour modelling, dataseries, Frequent Correlated Trends

1. Introduction

In recent years, behavioral sciences have received a lot of attention from the informatics perspective. This fact is mainly due to current demands for behavior analysis and understanding outstripping the capability of traditional methods and techniques in behavioral sciences. New computational tools for representing and working with behaviors are very welcomed and

*Corresponding author

Email addresses: mdelgado@ugr.es (Miguel Delgado), aragorn@ugr.es (Waldo Fajardo), miguelmolina@ugr.es (Miguel Molina-Solana)

[1]Department of Computer Science and AI. ETSIIT. c/ Daniel Saucedo Aranda, s/n 18071 Granada, Spain

Preprint submitted to Information Sciences *February 15, 2012*

a growing field of research, namely *Behavior Informatics* [5], is receiving increasing recognition.

Intuitively, we can define a behavior as a set of actions that are characteristic of one individual or phenomenon. These actions are ordered (or partially ordered) in some way, and indexed by a variable, which is generally *time*. By representing behaviors, two goals are aimed: identification and tagging of the behaviors, and forecasting future actions within them.

Behaviors, as we have just defined them, can be found in many different domains. The following are some illustrative examples of such phenomena that can be represented and individually identified (some of them will be later described in Section 5):

- The weather in a given area, represented as a series of observations at different time instants, including information such as temperature, precipitations or wind speed [12].
- The movements of a (injured) knee when doing some rehabilitation exercises, by monitoring the position of several reference points at different time instants [2].
- The way of playing an instrument. A particular performance of a piece of music can be represented as a series of notes with its respective duration and volume, among others attributes [15].
- The way a human being behaves within Ambient Assisted Living, with the aim of identifying strange actions and situations of potential danger [6].
- The personalization of mobile services. As mobile devices increase their capacity, new services and applications are developed which need modeling the user behavior and context [3].
- The interactions between currency exchanges. Several works have studied how several currencies behave against each other at different financial situations [25].

These real-world phenomena can be naturally represented by data series. As databases from most of industrial and biological areas often contain timestamped or ordered records, data series are gaining weight as a suitable source of information, and working with them has become an important machine learning task. Those records are generally obtained in an automatic manner from different sensors.

Two main goals of data series analysis are found in literature [4]: forecasting and modeling. The aim of forecasting is to accurately predict the next values of the series, whereas modeling aims to describe the whole series. Even though they can be sometimes related, they usually differ as finding a proper model for the long-term evolution might not be the best approach to predicting the short-term evolution and viceversa.

Forecasting and modeling are also the main tasks concerning behaviors, as we previously said. Therefore, data series are a suitable representation for behaviors, being also the most common one.

In either case, and whatever the goal of a particular data series analysis is, data representation is a crucial task anyway. It is hence required a formal representation capable of modeling the complexity of the particular data. This representation must be more reduced than representing all the observations of the phenomenon, but still describe it accurately enough.

An additional problem is that information is hardly certain, complete and precise; more on the contrary, it is usually incomplete, imprecise, vague, fragmentary, not fully reliable, contradictory, or imperfect in some other way. Historically, two ways of addressing imperfection have been employed for representing information in a computer [16]:

- The first solution consists in restricting the model to only that part of the available information of the real world that is accurate and reliable. Such a constrained approach avoids further complications of representation, but lacks the capacity of capturing the whole rich notion of information in human cognition and is generally very limited.

- The second solution implies developing models capable of representing imperfect information. As this approach allows a greater number of applications, it is the one that developers usually implement in their systems. However, those models cannot successfully cope with the whole range of imperfections that generally appear in real life, and in many occasions data are simplified to a point that makes them easily treatable with current computational tools, but losing part of their meaning.

Due to this lack of general systems capable of dealing with any kind of imperfect data, developers have been forced to handle this information in an ad-hoc manner; that is, by devising specific algorithms and systems for each new application, domain and representation. Therefore, in order to model the real world as accurately as possible, several approaches for dealing with imperfect information have been introduced and studied.

Although some schemes have been proposed for directly handling imperfect information coming as data series [13, 18], most of the research has focused on similarity measures to deal with imperfection. Hence there is still a need for further research and new practical systems capable of accurately modeling imperfect data series, and the field of *Behavior Informatics* will be greatly benefit by such advances.

This paper addresses this necessity by proposing a novel approach for representing imperfect behaviors —concretely uncertain and imprecise— that come in the form of multivariate data series. Our work shows how we can represent underlying local trends in the data in an easy and effective way, without a complicated formalism. Some other works have identified the necessity of focusing in frequent local cues for behavior modeling [8, 17]. The further aim of the work is addressing the issue of soft data series[2] recognition and comparison.

Specifically, our proposal identifies given behaviors through capturing their general footprint by means of discovering repetitive patterns in one dimension and their interdependences with patterns in other dimensions. This process can be divided in the following three stages:

1. a high-level abstraction of the observations within each dimension;
2. a tagging according to the patterns identified in one of the dimensions;
3. characterization of behaviors as sets of frequency distributions.

Most proposals for representing behaviors are domain-specific, being designed for the particular problem they are applied to [12, 21]. Even though they perform well when evaluated, these models cannot be easily applied to different domains without effort and fundamental modifications. Our proposal, like [9], is intended as a general framework that can be applied to several domains.

Two additional advantages of our method are of special relevance. In first place, the representation of the behavior is finite and constant in size for a given problem, regardless of the number of observations. This fact contrasts

[2]The term *soft data series* is used for series whose values are not accurate or verifiable.

with other works [17, 21] in which the size of the representation completely depends on the amount of data available.

The second advantage is an incremental representation which can be calculated on-line very easily: when a new value is observed, this information is included in the representation, which is immediately updated. That is not the case in other proposals [25, 21, 7] which need to recalculate the whole representation when new observations are available.

These two features capacitate the method to automatically offer a representation of the data series until any given observation, and allow it to deal with data series of infinite length. Our proposal is specially aimed to those behaviors with a large number of observations.

The rest of the paper is organized as follows. Section 2 offers an introduction to data series. Section 3 is intended as a general introduction to imperfect data, describing the main forms of imperfection and the problems to address. In Section 4, we describe the proposed model, *Frequent Correlated Trends*, and the developed system, including data gathering, representation and distance measurement. This section also introduces the formal notation used and an illustrative example. In Section 5, some potential domains of application are proposed, and the use of the model on them is described. The paper concludes with final considerations and pointing out future work in Section 6.

2. Data series

As data series[3] have an increasing popularity and they are the formalism we will use in this paper, we devote this section to briefly introduce them. The interested reader should refer to any basic reference on time series analysis (for instance, [4]) for further information about ordered series.

Databases in areas such as Engineering, Medicine or Finances often contain timestamped or ordered records. The analysis of data series (and time series in particular) is then of great interest in these areas, and searching for similarities between data series is fundamental for several data mining tasks

[3]The reader will notice that in the following we mainly focus our descriptions on time series. That is due to the fact that *time* is by far the most common indexing variable in data series. However, we will keep using the term *data series* to make explicit that our model is not limited to any particular kind of series.

(e.g. classification, rule gathering, clustering or finding patterns) within these domains.

A data series A can be intuitively defined as an ordered sequence (finite or not) of values obtained at successive intervals of an indexing variable —often *time*. Each one of these observations A_i takes values from a domain $\mathcal{U}^n$.

According to the value of n, we have several kinds of series. If $n = 1$, each element is a single (scalar) value and we have an *univariable* series. On the other hand, if $n > 1$, each element is a vector with n components and the data series is a *multivariable* one. We can distinguish two kinds of multivariable series. A *multidimensional* series is a series in which the majority of variables are independent. A *multivariate* series is a series that has many dependent variables that are correlated to each other to varying degrees.

In many data series, data follow recurring seasonal patterns. We say that a data series has serial dependence if the value at some point A_i is statistically dependent on the value at another point A_j. However, in many situations, those recurring patterns can occur at arbitrary points, because they are indexed by events happening in other series. In other words, values in series A^r (the series with the r-th component of each value of A) could be dependent of those of series A^s. That is the case in multivariate series. Our hypothesis is that the correlation among patterns of different dimensions of the same series is an interesting source of information should we want to summarize the series.

Whilst data series can be infinite in length, computers can only deal with a finite number of pieces of information. Therefore, in order to represent an arbitrary data series in a computer, it is mandatory to employ a suitable representation of those data, reducing their size. This representation is then approximate and implies a trade-off between accurately capturing the data and representing it in the finite memory of a computer.

Such series of data which are composed of a huge amount of values are called data streams and they open a new area of research in which the tasks of classification, clustering, indexing and mining of series of data become more challenging. In this context, observations have to be processed in real-time and cannot be permanently stored for further query. Even more, the observations might evolve over time.

Due to the characteristics of our learning algorithm (see Section 4.2), *Frequent Correlated Trends* could be applied to data streams with the only addition of a proper module to deal with that evolution. In fact, some algorithms with the same fundamental idea of ours (i.e. using a reduced representation that can be calculated in real-time) has already been proposed for data streams [1].

3. The problem of Imperfect data

Weaver suggested in [20] that several scientific problems are not solvable by a simple formula or probability theory. In fact, what really underlies on his claim is a more fundamental statement: 'almost all real-life problems cannot be solved by conventional (precise) mathematics'.

Not in vain, we experience that information in most domains is usually incomplete, imprecise, vague, fragmentary, not fully reliable, contradictory, or imperfect in some other way. Imperfect information might result from using unreliable information sources, it can be the unavoidable result of information gathering methods that require estimation or judgment, or be produced by a restricted representation model.

In general, these various information deficiencies may result in different types of imperfection. According to [16], inconsistency, imprecision, vagueness, uncertainty, and ambiguity are the five basic kinds of imperfect information.

We say that we have *inconsistency* when one aspect of the real world is irreconcilably represented more than once in a data set. For example, having both '27' and '28' as values for John's age. Information inconsistency is a kind of semantic conflict that usually arises when integrating information from several sources.

Imprecision and *vagueness* are both related with the impossibility to give a concrete value to an element. The correct value is within a range of values, but there is no way of knowing which one to choose. For example, 'between 100 and 120 kilograms' and 'very heavy' for John's weight are imprecise and vague values respectively. Vague information is usually represented by linguistic terms.

Uncertainty indicates the degree of truth of a value. It expresses how sure one can be about a statement. 'It is almost sure that John is his brother' is an example of information uncertainty.

When an element of the model can have several possible interpretations, we say it is *ambiguous*. In general, if values are not accompanied by their units, it is impossible to say if a figure is high or low. A length of 1000 is meaningless unless it is stated if those are millimeters or kilometers.

Generally, several different kinds of imperfect information can coexist with respect to the same piece of information. In many real-world problems we have (or could have) statements like the following: ‘it is α-certain that X is A’, being X a variable, α a certainty degree, and A an imprecise value. In a statement like ‘it is almost sure that John is a nice person’ two sources of imperfection are present: uncertainty (for ’almost sure’) and imprecision (for ’nice person’).

During the last century, there has been a great shift in the way of dealing with imperfection. For many years, it was considered undesirable in science and it was avoided at all means. However, at one point, scientists assumed that imperfection is not only an unavoidable reality, but it is, in fact, a useful source of information.

We are now within this second school of thought, which sees imperfection as relevant. Because of that, current solutions to problems use, in one way or another, suitable tools for dealing with and representing imperfection.

3.1. Imprecision and uncertainty

Imprecision and uncertainty are the most common kinds of imperfection found in data. That is especially true when dealing with problems related with biological systems. Because of this fact, we will devote a few extra lines to these two specific kinds of information imperfection.

As indicated before, imprecision and uncertainty state for different things. To further illustrate the differences between them, we reproduce here the example from Klir and Yuan’s book [11, page 177].

> “Consider the jury members for a criminal trial who are uncertain about the guilt of innocence of the defendant. The uncertainty seems to be of a different type; the set of people who are guilty of the crime and the set of innocent people are assumed to have very distinct boundaries. The concern, therefore, is not with the degree to which the defendant is guilty, but with the degree to which the evidence proves this membership in either the crisp set of guilty people or the crisp set of innocent people. We assume that perfect evidence would point to full membership in one and only one of these sets. However, our evidence, is rarely, if ever, perfect, and some uncertainty usually prevails.”

Uncertainty is the imperfection described here, and it arises from not knowing if an element belongs in a set or to another. On the contrary, imprecision comes from not knowing the degree of an element belonging in a set (i.e. the lack of sharp boundaries).

Several data models have been proposed to handle uncertainty and imprecision, and most of them are based on the same paradigms. Imprecision is generally modeled with fuzzy sets, and uncertainty with fuzzy measures.

The theory of fuzzy sets, introduced by Lotfi A. Zadeh in 1965 [23], is an extension of the notion of classical sets, allowing elements to have degrees of membership, not just the binary terms of belonging and not-belonging. As said, it has been widely used to represent imprecision.

Uncertainty, on the other hand, is generally addressed by the theory of fuzzy measure, which indicates the degree of evidence or certainty that a particular element belongs in the set. Three special cases of fuzzy measure theory have been widely studied: evidence theory, possibility theory and probability theory. Historically, the last one has been by far the most employed in literature.

The approach we will describe in the next section lies within this general framework —dealing with imprecision by means of fuzzy sets, and with uncertainty by employing probability. This representation is general enough to allow its application to several problems and domains, and can be easily understood and implemented.

The work by Kriegler and Held [12], while solely focused on dealing with global warming, presents another alternative to deal with both uncertainty and probability by means of p-boxes and belief functions. However, that models lack the generalization to any domain and the simplicity that our work aims to.

We can also find in the literature the concept of the probability of a fuzzy event. It was Zadeh [24] who proposed it. Yager has also addressed the issue in [22] proposing a new definition of this probability. However, the problem of establishing a theory of probability for fuzzy events still remains an open one [19].

Because of this lack of well-definition we do not follow this approach in the present paper. We aim for a well-defined framework of representation that fuzzy probabilities cannot yet offer. However, we hope this formalism to become a solid one in the near future, providing a new tool for representing both imprecision and uncertainty simultaneously.

4. Our proposal: Frequent Correlated Trends

So far in this paper we have explained the kind of problem we want to address, and have introduced the concepts of data series and imperfect information. In this section, we propose and describe a new model for representing imperfect behaviors —concretely uncertain and imprecise— that come in the form of multivariate data series. That kind of behaviors are widely found in real-life problems. The model represents underlying local trends in the data in an easy and effective way, without a complicated formalism. We also describe the corresponding learning algorithm.

The representation presented in this paper, *Frequent Correlated Trends*, is related with Concept learning [14, chapter 2] in the sense that both aims to achieve a minimum representation for a concept, in our case, a behavior. Both formalisms try to summarize a particular behavior, simplifying its original representation but saving the specific attributes that differentiates that behavior from any other.

In order to apply our model and algorithm, the following three hypothesis with respect to the data are assumed:

- The series of data contain one or several recurrent patterns. Those patterns will be the ones we will try to identify.
- If a certain pattern was observed in the past, it will eventually happen again in the future.
- The data series are long enough for the information obtained from them to be representative. The longer the data series is, the more accurate the representation.

4.1. Representation model

The mathematical model of our proposal is as follows. First, let T be an ordered set. We define a *behavior* X as a multivariate data series with observations at points of T. Each observation $x_i \in X$ (at a point $i \in T$) is a vector of n scalars that takes values from a domain $\mathcal{U}^n$. Each component $d \in D$, with $|D| = n$, is considered a dimension or *point of view* of the behavior X; X^d is the univariate series[4] of component d, and its domain is defined by the d-th component of $\mathcal{U}^n$.

[4]Note that a multivariate data series can be seen as a set of several univariate data series with values at the same points of an indexing variable.

In an ideal situation, we would have very precise measurements for each x_i, and any repetition X' of a behavior X would be identical. In practice, it is hardly the case. We cannot fully rely on the observations as their accuracy is not guaranteed. In fact, what we usually have are approximated series of values derived from imprecise measurements. Nevertheless, there are some commonalities between several instances of behaviors, as we are able to identify repetitions of them.

Because most behaviors are inherently imprecise and changing, it is mandatory to employ a representation model capable of handling these imperfections, as we have seen in former sections. This fact does not imply any constraints. More on the contrary, it appraises the general case which is actually the most common one.

Therefore, as several repetitions of the same pattern are not identical, it is not feasible to learn a deep model about a behavior, i.e. to learn rules mapping specific patterns to each behavior. The alternative we propose is to take a more global perspective of a behavior by trying to capture its essential (and recurrent) patterns. To deal with the imprecision in the observations, we propose to use a contour segmentation as a way to capture the structure of a behavior. This is also the alternative that other researchers take [8, 10].

Our approach for dealing with the identification of behaviors is based on the acquisition of *frequent correlated trends* that characterize each particular behavior to be identified. Specifically, these trends model the relationships between structural patterns in a reference dimension, and patterns in other dimensions. Notice that correlated trends are not trying to characterize the behavior with respect to an expected global value.

It is important to note here that we focus on the computational representation of the observations that came from several sensors, regardless of which ones they are. Therefore, our approach is independent of the technologies used to collect the information.

For illustrative purposes we will present a concrete example about weather (see Section5.2), along with the formal description, in order to clarify concepts and procedures regarding the method and model. In particular, we will focus on the weather in a given area, which can be understood as a behavior. One of the most common ways of representing these data is by means of a series of observations at different time instants ($T = time$). The model we propose in this paper is indeed appropriate for modeling this phenomenon because the exact values are not really important and the repetitions are not exactly the same.

Therefore, the behavior X will be the weather in a given area, and it is described as a set of observations $x_i, i \in \{1 \cdots 15\}$. Each observation x_i has three components corresponding to three ($n = 3$) aspects of the weather at that point i. The set of those aspects is $D = \{$*temperature, precipitations, wind speed*$\}$, and the concrete values of the observations x_i^d for each dimension $d \in D$ are the following[5]:

$$\begin{aligned}
X^{temperature} &= \{19, 21, 21, 25, 23, 22, 19, 17, 21, 22, 22, 18, 19, 22, 24\}\\
X^{precipitations} &= \{98, 102, 110, 93, 95, 71, 67, 52, 63, 75, 80, 95, 72, 104, 98\}\\
X^{windspeed} &= \{7, 6, 1, 3, 4, 7, 11, 15, 14, 7, 9, 11, 9, 9, 7\}
\end{aligned}$$

4.2. Learning algorithm

In order to transform real data to the frequent correlated trends formalism, a learning algorithm is necessary. In the rest of this section, we describe the process and discuss how we do such transformation. Algorithm 1 summarizes the process.

As previously said, our representation aims to describe a behavior X in terms of the relationships between patterns from a reference dimension $I \in D$ and patterns from the rest of dimensions of X. Patterns in X^I will be used to index repeating patterns in other dimensions. In other words, patterns in series $X^d, d \neq I$ will be related with those in X^I.

In most behaviors, the selection of I is clear by the semantics of the domain and does not represent a problem. In the case of not knowing which series is appropriate to use as reference, the correlations between dimensions should be studied and the dimension with a greater number of correlations with the rest should be selected. This way, we tend to maximize the amount of correlated patterns. In fact, this might very well be the way to proceed in all cases; however, using the semantics can clarify the representation in many domains.

It should be noted here that, although we have defined X as being indexed by T, this dimension is not a choice for I, because T is an ordered set and thus no patterns can arise from its consecutive values, apart from the monotone growth. Also, $T \notin D$.

[5]As our method is aimed for behaviors with a large number of observations, it will not be able to get any meaningful information from these data. However, as an illustrative example of the process, we will employ this reduced number.

Algorithm 1 General Learning algorithm for Frequent Correlated Trends

Definitions:

$T \subset \mathbb{R}$ is an ordered set
X is a behavior described as a multivariate series indexed by T
D is the set of dimensions of X
n is the cardinality of D
X^d is the univariate series with only observations from dimension d
s is the number of observations for each segment
$\mathcal{P}_s$ is the set of indexing patterns of size s
$\mathcal{Q}_s$ is the set of qualitative shapes size s

Algorithm:

```
Select a reference dimension I ∈ D
Tag each interval in X^I with a value from {a, d, e}, obtaining X'^I
Segment X'^I in groups of size s
Tag each of those segments with an element from P_s, obtaining 𝒳^I
for all dimensions d ∈ D, d ≠ I do
  Calculate X̄^d as the average value of observations of X^d
  Transform each observation x_i^d ∈ X^d as follows:
  if x_i^d ≥ X̄^d then
    x'_i^d ← +
  else
    x'_i^d ← −
  end if
  Segment X'^d in groups of size s
  Tag each of those segments with an element from Q_s, obtaining 𝒳^d
  for all p ∈ P_s do
    Count the number of simultaneous occurrences of p in 𝒳^I and q ∈ Q_s
    in 𝒳^d
    Build a probability distribution for Q_s given p from those values
  end for
end for
```

The next step consists of transforming the reference series X^I using a simple contour criteria. Two consecutive (imperfect) observations form an interval. If we consider just the directions of the intervals, three options are available: (a)scending, (d)escending and (e)qual. Each interval in the reference series X^I is then tagged with one of these three options.

At this point, we have to decide the size s of patterns. Small patterns are preferred because they are more easily found in the series. The longer the pattern is, the more difficult it is to find occurrences. We define the set of indexing patterns of size s, $\mathcal{P}_s$, as the set of all permutations with repetitions of size $s-1$ of elements from $\{a, d, e\}$. If groups of three values (i.e. two intervals) are considered, nine different patterns arise, $\mathcal{P}_3 = \{aa, ad, ae, da, dd, de, ea, ed, ee\}$.

We then segment the indexing series X^I in groups of size s and tag each group with the corresponding indexing pattern from $\mathcal{P}_s$. We will obtain a new series $\mathcal{X}^I$. To do so, we propose to use a sliding window of s consecutive observations that moves in steps of one.

Following with our example, we will select *temperature* as the indexing dimension I, because we would like to relate patterns in dimensions *precipitations* and *wind speed* with those in *temperature*. We aim to obtain relations such as: "65% of the times, when the temperature rises in two consecutive days, the second day it rains more than the first". Once selected, the segmentation of that series will take place as formerly described, obtaining $\mathcal{X}^I$. In summary, $X^{temperature}$ is transformed into X'^I and after into $\mathcal{X}^I$:

$$\begin{aligned} X^{temperature} &= \{19, 21, 21, 25, 23, 22, 19, 17, 21, 22, 22, 18, 19, 22, 24\} \\ X'^I &= \{a, e, a, d, d, d, d, a, a, e, d, a, a, a\} \\ \mathcal{X}^I &= \{ae, ea, ad, dd, dd, dd, da, aa, ae, ed, da, aa, aa\} \end{aligned}$$

Similarly, the rest of data series $X^d (d \in D, d \neq I)$ conforming the behavior X are transformed by employing a qualitative binary transformation in the following way: each value x_i^d is compared to the average value $\bar{X}^d$ in the behavior, and it is transformed into a qualitative value where $+$ means 'the value is higher than or equal[6] to the average', and $-$ means 'the value is lower than the average'.

[6]In our current model, we have arbitrarily decided to include equal values in the + set. However, another qualitative value, namely =, could be used to distinguish between those values that are greater than the average and those which are equal (or almost equal).

Note here that if the behavior can be described by a finite series, the average value (a vector with the average values for each component) can be easily computed with the usual formula:

$$\bar{X} = 1/N \cdot \sum_{1 \le i \le N} x_i$$

Otherwise, the average can be calculated incrementally. However, an initial estimation of it (by any means) would be required. In this situation the average value is different depending on the position of the observation been calculated, but tends to become stable as more values are computed.

In our example, we will proceed to transform series *precipitations* and *wind speed*. To do so, we first calculate the average value for those series using all available observations. Those average values are 85 mm and 8 mph, for precipitations and wind speed respectively. We will then compare all values in $x^{precipitations}$ with 85 and transform them to $+$ if the value is greater than or equal to 85, and to $-$ if lower. We do the same process with $x^{windspeed}$, obtaining the following series:

$$\begin{aligned}
X^{precipitations} &= \{98, 102, 110, 93, 95, 71, 67, 52, 63, 75, 80, 95, 72, 104, 98\} \\
X'^{precipitations} &= \{+, +, +, +, +, -, -, -, -, -, -, +, -, +, +\} \\
X^{windspeed} &= \{7, 6, 1, 3, 4, 7, 11, 15, 14, 7, 9, 11, 9, 9, 7\} \\
X'^{windspeed} &= \{-, -, -, -, -, -, +, +, +, -, +, +, +, +, -\}
\end{aligned}$$

Being s the size of the segment and r the number of different qualitative values, there are r^s possible resulting shapes. In the current example, since we are segmenting the series in groups of three observations and using two qualitative values, eight (2^3) different patterns may arise. We note these possibilities as $\mathcal{Q}_3 = \{---, --+, -+-, -++, +--, +-+, ++-, +++\}$. We then segment the *precipitations* and *wind speed* series in groups of three observations and tag these groups with patterns in $\mathcal{Q}_3$, obtaining $\mathcal{X}^{precipitations}$ and $\mathcal{X}^{windspeed}$ respectively:

$$\begin{aligned}
\mathcal{X}^{precipitations} &= \{+++, +++, +++, ++-, +--, ---, ---, \\
&\quad ---, ---, --+, -+-, +-+, -++\} \\
\mathcal{X}^{windspeed} &= \{---, ---, ---, ---, --+, -++, +++, \\
&\quad ++-, +-+, -++, +++, +++, ++-\}
\end{aligned}$$

Once we have transformed and tagged all the series, we can represent them with the frequent correlated trend formalism. A frequent correlated trend for a given point of view $d \in D$ of a behavior X is represented by a set of discrete frequency distributions for that given dimension d. Each of these frequency distributions represents the way patterns in $\mathcal{X}^d$ behaves with respect to indexing patterns in $\mathcal{P}_s$. In other words, we have a frequency distribution for each element in $\mathcal{P}_s$.

To populate the representation of a particular behavior X and dimension d, we take advantage of the tagging already performed over the observations. We construct a histogram for each indexing pattern $p \in \mathcal{P}_s$. Histograms have a bin for each element $q \in \mathcal{Q}_s$, and are constructed by calculating the percentage of points $i \in T$ where $\mathcal{X}_i^I = p$ and $\mathcal{X}_i^d = q$. These histograms can be understood as discrete probability distributions of $\mathcal{Q}^s$ given p. Thus, frequent correlated trends capture statistical information of how a certain phenomenon behaves. Combining frequent correlated trends from different dimensions of a behavior, we improve its representation.

Figure 1 shows, for a given dimension d and indexing pattern p, the discrete probability distribution obtained from four behaviors. We can see that there are eight bins, corresponding to the elements in $\mathcal{Q}_3$. The figure indicates how different behaviors tend to behave against the pattern p.

We can observe that samples are not equally distributed. For instance, *Behavior 1* has a higher propensity to extend the values (see $+++$) while an opposite behavior can be observed for *Behavior 2* (see its values for the two left qualitative shapes).

If Figure 1 were from our weather example, it could show how the weather in four different locations behaves. In particular, it could show for a pattern p in *temperature* (let's say consecutive increments in temperature: $p = aa, d = \mathit{temperature}$), how a certain aspect of the weather (for instance, *precipitations*) statistically behaves. Therefore, at location 1, consecutive increases in temperature would imply increasing precipitations; whereas at location 2, an opposite behavior would be the norm.

4.3. Computing distance among behaviors

In many occasions, the final goal of representing behaviors is their posterior identification and classification. If that is the case, behaviors —which we describe as a collection of frequent correlated trends— are used as the patterns to compare with when a new instance is presented to the system. Each behavior can then be viewed as a point in a space of dimensionality

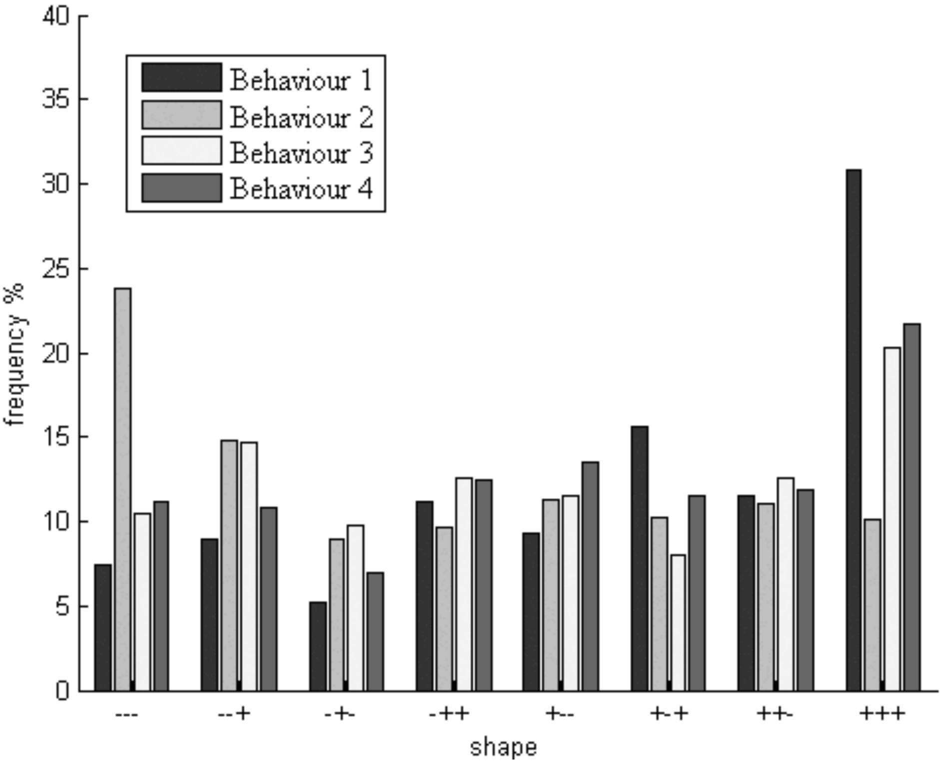


Figure 1: Example of the representation of four different behaviors for a given dimension d and indexing pattern p. Each bar shows the % of occurrence of the corresponding shape in dimension d, within all occurrences of p in the whole behavior. In terms of our weather example, this figure could show how precipitations behave against a given pattern of temperatures (let's say two consecutive days of rising temperatures). We could see then that in *Behavior 1* there is a higher probability of more rain than usual, whereas in *Behavior 2* we have the opposite behavior.

$n \cdot |\mathcal{P}_s| \cdot |\mathcal{Q}_s|$, where n is the number of dimensions in the behavior; $|\mathcal{P}_s|$ is the cardinality of the set of indexing patterns; and $|\mathcal{Q}_s|$ is the cardinality of the set of qualitative shapes for representing the rest of series. In that case, the usual hypothesis is that similar patterns tend to be close in the representation space.

As the reader can observe, this representation tends to have a high dimensionality. This fact can be a source of problems if the number of samples is not large enough. Due to the so-called *curse dimensionality* (that is, the exponential growth of the space as a function of dimensionality), multivariate spaces of increasing dimensionality tend to be sparse. For that reason, if frequent correlated trends are to be used for learning and classification, it is mandatory to study the dimensionality of each concrete problem and the number of available samples. If these samples are not enough, it might be necessary to reduce the dimensionality of the model by means of any feature extraction mechanism.

In any case, to compare different behaviors, a distance measure is required. As an illustrative example, we propose in this paper a simple one based on Manhattan distance. It is described in the following lines.

The distance d_{XY} between two behaviors X and Y (represented with the frequent correlated trend formalism), is defined as the weighted sum of distances between the frequency distributions of each structural pattern:

$$d_{XY} = \sum_{p \in \mathcal{P}} w^p_{XY} \cdot dist(p_X, p_Y) \tag{1}$$

where $\mathcal{P}$ is the set of the different structural patterns considered; $dist(p_X, p_Y)$ measures the distance between two frequency distributions (see (3) below); and w^p_{XY} are the weights assigned to each pattern. Weights have been introduced for balancing the importance of patterns with respect to the number of times they appear. Frequent patterns in the indexing series are considered more informative due to the fact that they come from more representative samples. Weights are defined as the mean of cardinalities of respective histograms for a given pattern p:

$$w^p_{XY} = (N^p_X + N^p_Y)/2 \tag{2}$$

Mean value is used instead of just one of the cardinalities to assure a symmetric distance measure in which w^p_{XY} is equal to w^p_{YX}. Cardinalities could be different because the data series of different instances of a behavior could have different cardinality.

Finally, distance between two frequency distributions is calculated by measuring the absolute distances between respective patterns:

$$dist(u, v) = \sum_{q \in \mathcal{Q}} |u_q - v_q| \tag{3}$$

where u and v are two frequency distributions for the same pattern; and $\mathcal{Q}$ is the set of all possible values they can take.

When several dimensions (i.e. points of view of a behavior) are considered, we propose to simply aggregate the individual corresponding distances.

4.4. Computational complexity

As previously indicated, our formalism requires a finite amount of memory to represent a behavior described by any multivariate data series. This is an important fact because data series tend to be very long and often infinite.

In particular, the complexity in memory of our representation is proportional to $n \cdot |\mathcal{P}_s| \cdot |\mathcal{Q}_s|$, where n is the number of dimensions in the behavior; $|\mathcal{P}_s|$ is the cardinality of the set of indexing patterns; and $|\mathcal{Q}_s|$ is the cardinality of the set of qualitative shapes for representing the rest of series. This size is fixed for each problem and it is totally independent of the length of the series.

Following with our example, $n = 2$, $s = 3$, $|\mathcal{P}_3| = 9$ and $|\mathcal{Q}_3| = 8$. Therefore, a behavior in our weather example would be a point in a space of dimensionality 144, the result of $2 \cdot 9 \cdot 8$.

As indicated in the introduction, building the representation has linear complexity in time. Not in vain, we can tag and transform values as they appear —or with a minimum delay due to the size of the sliding window. As the formalism of frequent correlated trends corresponds to sets of discrete probability distributions, they can be populated incrementally without any problem.

5. Potential domains of application

The model (*Frequent Correlated Trends*) that we have described so far in this paper, for representing behaviors that come as imprecise and uncertain data series, can be used in several domains. We will describe three of them to exemplify the possibilities, but many others may be suggested (see the Introduction).

5.1. Music Performance

It is a fact that, in music performances, musicians do not play exactly what is written in the score. They deviate from it and enrich the performances with tempo, dynamics and tuning deviations. These small deviations are not always identical, neither exact, and they are not explicitly encoded in the score. Even when a performer try to play several times the same way, they cannot exactly achieve it.

Research on expressive music performance has traditionally focused on the analysis of score representation. Nevertheless, the advances on audio content analysis and description have enabled the study of music performance by directly analyzing audio recordings of professional musicians. But due to the fact that the transformation from audio signal to symbolic representation is not accurate (current feature extraction tools are far from being perfect), some unavoidable imperfection is added to the information. In other words, the measurements are no longer reliable and we need a representation capable of dealing with this fact.

From a computational point of view, a music performance can be described as a series of notes (events). Each one of these notes can be defined by several parameters such as length and volume. Therefore, a music performance can be expressed as series of (imprecise) observations in several dimensions, which is the kind of representation that our model requires as input.

The way of playing a musical piece is then a problem in which *Frequent Correlated Trends* can be applied. In fact, that was already done in [15] showing that good results can be achieved despite very limited information.

In that study, we departed from audio recordings of music performances. These recordings were transformed into sequences of notes with pitch, duration and energy attributes. Therefore, three data series (pitch, duration and energy) were used to represent each performance. The pitch series was used as an index for the other two. A melodic contour criterion was used as a way to identify the basic shapes of 3 notes within those series. In other words, we located within the pitch series the nine possible shapes that can be produced with combinations of ascending, descending and null intervals.

Most works in performer identification rely on annotated information or a score. However, our approach only requires an audio signal, because frequent correlated trends are able to deal with all the imprecision and uncertainty inherent to those data.

The feasibility of this representation was shown for a dataset of monophonic violin recordings from 23 well-known performers. Given an audio recording of a musical piece as input, the system was able to recognize the violinist playing that piece better than chance level and, probably, human experts. In particular, the system achieved a success rate of 20% in the worst case (outperforming the 4.3% of a random classifier), and higher than 50% if 7 trials were allowed (random classifier out of 23 possible answers would only get 30%). The interested reader should refer to [15] for more information on those experiments and results.

5.2. Weather

Modeling and forecasting the weather in an area is far from being a new problem. For many years scientists have tried to build better representation models for the weather with the aim of providing new tools to predict the weather in the following days, weeks or years [12]. The importance of accurate predictions is based on the great socio-economic impact they have.

We can understand the weather in a given area as a behavior that has a particular set of characteristics. These allow us to distinguish among weathers at different locations. One straightforward way of representing such information is by means of data series. In that case, the weather in an area can be described as a set of measurements from several variables such as temperature, air pressure, relative humidity, precipitations, wind speed, and so on.

We propose to use our *Frequent Correlated Trend* model to represent the weather in an area during a long period. The current representation of weather as multiple series of data can be used to populate our model, and we have already shown how by means of the example in this paper.

Frequent Correlated Trends would aim to find relations between variables such as the followings: "65% of the times, when the temperature rises in two consecutive days, the second day it rains more than the first", or "when the temperature is stable for two days and the first day is sunny, the second day will be sunny 80% of the times".

5.3. Rehabilitation

Medicine has historically benefited from new technologies and advances on the field of Computer Science. Not in vain, many researchers have found in Medicine a great area of application for their new developments, being the outcomes of this synergy generally huge and of great importance for the people.

In particular, one area in which computers can be applied with great success is rehabilitation. As monitoring is an essential task, a great deal of observations is available. A better comprehension of medical data is a must for improving medical effectiveness, and good data models to represent certain injuries are also welcome. Databases and data-mining allow to handle all that information in easy ways and can provide additional tools to interpret those data.

Doctors collect many data from patients in order to assess whether an injured joint (e.g. a knee) is currently fine, or need further rehabilitation. If repeated patterns between graphs belonging to different patients with the same complaint are discovered, a proper representation model can then be used as an aid for early detecting such complaints. Therefore, detecting those patterns is a very useful task for monitoring injuries, discovering fraudulent sicknesses, and early detection of potential injuries [2].

We aim at using *Frequent Correlated Trends* to represent the evolution of an injured joint. We aim to be able to describe the movements the joint is doing, and classify them in different categories according to the degree of injury and the rehabilitation needed.

A joint can be monitored by means of different sensors, and several data series from its behavior can be obtained. Again, these series can be transformed into the representation we have described in this paper, providing a new approach for modeling the performance of an injured joint.

6. Conclusions

Many phenomena in the real world can be understood as behaviors, because they follow some rules and behave in characteristic ways. The computational representation and classification of them is a task of great interest for researchers in behavioral sciences as they are in continuous need of new tools and methods to represent, summarize and understand behaviors of growing complexity.

Data series are one of the representations with increasing importance lately. They are defined as an ordered sequence of data at given intervals of an indexing variable (e.g. time). Because of that, behaviors in many domains can be defined as imprecise and uncertain multivariate data series.

Although the problem of representing imperfect data has been addressed many times in the past, the lack of a general and universal solution obligates to build ad-hoc solutions for different problems. For that reason, there is still a need to find new solutions and models to represent such information.

Our proposal, which assumes that some kind of commonality exists among several instances of the same behavior in a given phenomenon, represents those imperfect data series as a set of probability distributions. To do so, it first transforms the imperfect observations into qualitative values. Then, it selects a dimension of the behavior and uses it to look for correlations with the rest of dimensions. These correlations are expressed as discrete probability distributions.

The main advantage of our method is that it employs a finite and constant representation in size for behaviors, regardless of their length. It allows for an incremental representation of the observations until a particular moment.

On the other hand, the main limitation of our model is that it is incapable of explaining the predictions it makes. Additionally, as the method is aimed for long series, its best performance is achieved when a large number of observations is available.

The feasibility of the model was shown in the domain of music performances. Experiments showed that this representation allows to identify violinists in a dataset of monophonic violin recordings from 23 well-known performers.

6.1. Further work

The first line of further work consists of applying the *Frequent Correlated Trend* model to different domains to test their feasibility. Some of them has been described here as illustrative examples, but many others can be candidates.

Besides that, we plan to try other fuzzy measures apart from probability theory (specifically possibility theory). Not in vain, both probability and possibility theories are suitable for modeling uncertainty, but each one excels in different types. As our framework aims to be a general one, it should address uncertainty in all their varieties. Obviously, for each problem it has to be decided which theory is more appropriate to represent the semantics of the observations.

A different issue would be to study the distance measure appropriate for each domain. We have defined in this paper a very simple one, but many others distance measures could be proposed to better account for the particular semantics of different applications.

Acknowledgments

The authors would like to thank the reviewers for their comments on earlier versions of this manuscript. Authors are supported by the Spanish Ministry of Education and Science (project TIN2009-14538-C02-01). Miguel Molina-Solana is also supported by FPU grant AP2007-02119.

References

[1] C. Aggarwal, A segment-based framework for modeling and mining data streams, Knowledge and Information Systems 30 (2012) 1–29.

[2] F. Alonso, A. López-Illescas, L. Martínez, C. Montes, J.P. Valente, Knowledge discovery using medical data mining, in: A. Colosimo, P. Sirabella, A. Giuliani (Eds.), Medical Data Analysis, volume 2526 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 2002, pp. 1–12.

[3] T. Bao, H. Cao, E. Chen, J. Tian, H. Xiong, An unsupervised approach to modeling personalized contexts of mobile users, in: Data Mining (ICDM), 2010 IEEE 10th International Conference on, pp. 38–47.

[4] P.J. Brockwell, R.A. Davis, Time Series: Theory and Methods, Springer, 1991.

[5] L. Cao, In-depth behavior understanding and use: The behavior informatics approach, Information Sciences 180 (2010) 3067–3085.

[6] M. Delgado, M. Ros, M.A. Vila, Correct behavior identification system in a tagged world, Expert Systems with Applications 36 (2009) 9899–9906.

[7] A. Georgakis, H. Li, User behavior modeling and content based speculative web page prefetching, Data & Knowledge Engineering 59 (2006) 770–788.

[8] G. Herbst, S.F. Bocklisch, Short-time prediction based on recognition of fuzzy time series patterns, in: E. Hüllermeier, R. Kruse, F. Hoffmann (Eds.), Computational Intelligence for Knowledge-Based Systems Design, volume 6178 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 2010, pp. 320–329.

[9] N. Johnson, D. Hogg, Representation and synthesis of behaviour using Gaussian mixtures, Image and Vision Computing 20 (2002) 889–894.

[10] S.W. Kim, J. Yoon, S. Park, J.I. Won, Shape-based retrieval in time-series databases, Journal of Systems and Software 79 (2006) 191–203.

[11] G.J. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice-Hall, 1995.

[12] E. Kriegler, H. Held, Utilizing belief functions for the estimation of future climate change, International Journal of Approximate Reasoning 39 (2005) 185–209.

[13] S.S. Liao, T.H. Tang, W.Y. Liu, Finding relevant sequences in time series containing crisp, interval, and fuzzy interval data, IEEE Transactions on Systems, Man, and Cybernetics 34 (2004) 2071–2079.

[14] T.M. Mitchell, Machine Learning, McGraw-Hill, 1997.

[15] M. Molina-Solana, J.L. Arcos, E. Gómez, Identifying violin performers by their expressive trends, Intelligent Data Analysis 14 (2010) 555–571.

[16] A. Motro, Sources of uncertainty, imprecision, and inconsistency in information systems, in: A. Motro, P. Smets (Eds.), Uncertainty Management in Information Systems: From Needs to Solutions, Kluwer Academic Publishers, 1996, pp. 9–34.

[17] B. Saleh, F. Masseglia, Discovering frequent behaviors: time is an essential element of the context, Knowledge and Information Systems 28 (2010) 311–331.

[18] Q. Song, R.P. Leland, B.S. Chissom, A new fuzzy time-series model of fuzzy number observations, Fuzzy Sets and Systems 73 (1995) 341–348.

[19] E. Trillas, T. Nakama, I. García-Honrado, Fuzzy probabilities: Tentative discussions on the mathematical concepts, in: E. Hüllermeier, R. Kruse, F. Hoffmann (Eds.), Computational Intelligence for Knowledge-Based Systems Design, volume 6178 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 2010, pp. 139–148.

[20] W. Weaver, Science and complexity, American Scientist 36 (1948) 536–544.

[21] W. Xu, L. Kuhnert, K. Foster, J. Bronlund, J. Potgieter, O. Diegel, Object-oriented knowledge representation and discovery of human chewing behaviours, Engineering Applications of Artificial Intelligence 20 (2007) 1000–1012.

[22] R. Yager, A note on probabilities of fuzzy events, Information Sciences 18 (1979) 113–129.

[23] L.A. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338–353.

[24] L.A. Zadeh, Probability measures of fuzzy events, Journal of Mathematical Analysis and Applications 23 (1968) 421–427.

[25] Y.Q. Zhang, X. Wan, Statistical fuzzy interval neural networks for currency exchange rate time series prediction, Applied Soft Computing 7 (2007) 1149–1156.

Identifying Violin Performers by their Expressive Trends

Molina-Solana, M., Arcos, J.L. & Gómez, E. (2010), "*Identifying Violin Performers by their Expressive Trends*", Intelligent Data Analysis. Vol. 14(5), pp. 555-571, DOI: 10.3233/IDA-2010-0439

- Status: **Published**
- Impact Factor (JCR 2010): 0.412
- Subject category:
  - COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE (98/108 Q4)

Intelligent Data Analysis 14 (2010) 555–571
DOI 10.3233/IDA-2010-0439
IOS Press

Identifying violin performers by their expressive trends

Miguel Molina-Solana[a,*], Josep Lluís Arcos[b] and Emilia Gomez[c]
[a]*Department of Computer Science and Artificial Intelligence, Universidad de Granada, Granada, Spain*
[b]*Artificial Intelligence Research Institute* (*IIIA*)*, Spanish National Research Council* (*CSIC*)*, Bellaterra, Spain*
[c]*Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain*

Abstract. Understanding the way performers use expressive resources of a given instrument to communicate with the audience is a challenging problem in the sound and music computing field. Working directly with commercial recordings is a good opportunity for tackling this implicit knowledge and studying well-known performers. The huge amount of information to be analyzed suggests the use of automatic techniques, which have to deal with imprecise analysis and manage the information in a broader perspective. This work presents a new approach, Trend-based modeling, for identifying professional performers in commercial recordings. Concretely, starting from automatically extracted descriptors provided by state-of-the-art tools, our approach performs a qualitative analysis of the detected trends for a given set of melodic patterns. The feasibility of our approach is shown for a dataset of monophonic violin recordings from 23 well-known performers.

1. Introduction

The advances in digital sound synthesis and computational power capabilities have allowed to provide real-time control of synthesized sounds. Expressive control becomes then a relevant area of research and a key challenge in the sound and music computing field [26].

According to Serra et al. [23], music performance is a complex activity that involves complementary facets from different areas such as acoustics, psychology and creativity. In this sense, the research in this field has a multidisciplinary character, ranging from studies that try to understand expressive performance to attempts at modeling performance aspects in a formal, quantitative and predictive way. One of the relevant research questions in this area is the modeling and identification of a given player or a playing style by analyzing a set of performances.

In this interdisciplinary research field, musical expressivity can be approached from different perspectives. One of them is the musicological analysis of music and the study of the different stylistic schools. This approach provides a valuable understanding about musical expressivity.

Another perspective related to the present article is the computational modeling of music performance analyzing music recordings. This analysis can be performed on a set of pieces specifically recorded for the intended study and related to the performance aspect we want to analyze, where specific expressive resources are emphasized. An alternative approach is to directly use commercial recordings for the

*Corresponding author: Miguel Molina-Solana, Department of Computer Science and AI, ETSIIT, c/. Daniel Saucedo Aranda, s/n 18071 Granada, Spain. Tel.: +34 958 240806; Fax: +34 958 243317; E-mail: miguelmolina@ugr.es.

1088-467X/10/$27.50

analysis of expressivity. This approach has several advantages: there are many recordings available (and some performers may have several ones); the performances are ‘real’ and gather the decisions taken by the performers without any external influence.

Nevertheless, working with commercial recordings has a drawback: these recordings do not come from a controlled scenario and the sound analysis may become more difficult. Our claim is that the advantages of working with commercial recordings overcome the drawbacks. Specifically, if a sufficient amount of data is available, we can take advantage of data-intensive techniques that both soft computing and machine learning fields provide.

In this paper, we focus on the task of identifying violinists from their playing style using descriptors that are automatically extracted from commercial audio recordings by means of state-of-the-art tools. First, since we consider audio excerpts from quite different sources, we assume a high heterogeneity in the recording conditions. Second, as state-of-the-art audio transcription and feature extraction tools are not 100% precise, we assume a partial accuracy in both the melodic transcription and feature extraction. In this research we are only dealing with monophonic violin recordings.

Taking into account these constraints, our proposal identifies violin performers through capturing their general expressive footprint with the following three stage process: (1) a higher-level abstraction of the automatic transcription that focuses on the melodic contour; (2) melodic segments tagged according to the E. Narmour’s Implication/Realization (IR) model [15]; and (3) the characterization of the way melodic patterns are played as a set of frequency distributions.

The rest of the paper is organized as follows: Section 2 describes related work on the field of expressive music performance and performer’s identification. In Section 3 we present the data collection being used. In Section 4, we describe the proposed *Trend-Based model* and the developed system, including data gathering, representation of recordings and distance measurement. In Section 5, some experiments to validate our approach are proposed and their results are described. The paper concludes with final considerations and pointing out future work in Section 6.

2. Related work

It is a clear fact that, in music performances, musicians do not play exactly what is written in the score. They deviate from the score and enrich the performance with tempo, dynamics and tuning deviations. There is an extensive literature on the analysis and modeling of music performances, focusing on different instruments and using different methodologies. Juslin et al. [11] propose a model that characterizes this variability in terms of four different sources: generative rules related to the musical structure; the emotional expression decided by the performer; random variations; and movement principles. In our research we are interested in capturing the variations due to the emotional expression and the performer’s analysis of a piece.

Lopez de Mantaras et al. [3] studied the expressiveness of some computer music systems based on artificial intelligence techniques and related it with the expressiveness of human-performed music. Research on expressive music performance has traditionally focused on the analysis of score representation. Nevertheless, the advances on audio content analysis and description (see [6] for a recent overview) allow the study of music performance by directly analyzing audio recordings of professional musicians. These performer’s variations have been modeled by applying machine learning techniques.

Related to the piano, Dovey [4] outlined an attempt to use inductive logic programming to determine various interpretative rules that pianist Sergei Rachmaninoff may have used during his pianoforte performances in an augmented piano. The goal was to determine general rules (in the form of universal predicates), concerning duration, tempo and key pressure, that underlie the way Rachmaninoff played the songs.

Relevant work on automatic piano performer identification has been carried out by the group led by Gerhard Widmer. Their research is based on acquiring a performance model via inductive learning and data mining techniques applied on a huge corpus of precisely manually measured performances. To cite some results, in [27] they applied an inductive rule learning algorithm to find general rules of music performance, while in [24] they propose a set of simple features for representing stylistic characteristics of piano music performers. In our approach, we do not require the availability of an annotated corpus. An interesting contribution is the paper by Goebl [5], where he focused on finding a 'standard performance' by exploring the consensus among different performers.

In a recent work, Saunders et al. [22] represent pianists' performances as strings with information related to changes in tempo and loudness. Finally, they use *Support Vector Machines* to identify the performer of new recordings.

Hong [10] investigated expressive timing and dynamics in recorded cello following an empirically perspective. He took the work by Todd [25] as a departure point, extending and testing it with some commercial recordings by famous cellists. Sapp's work [21] should also be cited as an interesting proposal for representing musical performances by means of scape plots based on tempo and loudness correlation.

Ramirez et al. [19] focused on saxophone recordings and studied deviations of parameters such as pitch, timing, amplitude and timbre, both at an inter-note level and at an intra-note level. They further extended the system to identify performers in Celtic violin recordings [20] and to the analysis of ornaments in bassoon recordings [18].

An alternative approach to inductive learning is the use of case-based reasoning (CBR) techniques. CBR is a lazy learning technique that, instead of building a general model from the existing examples, uses directly the examples to construct a solution for new problems. Examples of systems that deal with music performance using a CBR approach are SaxEx and TempoExpress systems. SaxEx [1] is a system capable of generating high-quality expressive solo performances of jazz saxophone ballads based on examples of human performers. TempoExpress [9] performs expressivity-aware tempo transformations of saxophone jazz recordings.

3. Musical data

We have chosen to work with Sonatas and Partitas for solo violin from J.S. Bach [13]. Sonatas and Partitas for solo Violin by J.S. Bach is a six work collection (three Sonatas and three Partitas) composed by the German musician. It is a well-known collection that almost every violinist plays during their artistic life. All of them have been recorded many times by several players. The reason of using this work collection is twofold: 1) we have the opportunity of testing our model with existing commercial recordings of the best known violin performers, and 2) we can constrain our research on monophonic music.

In our experiments, we have extended the musical collection presented in [17]. We analyzed music recordings from 23 different professional performers. Because these audio files were not recorded for our study, we have not interfered at all with players' style at the performance [12]. For the experimental results presented in this paper, we used three different movements: the Second and the Sixth movements of Partita No. 1 (both a *Double*) and the Fifth movement of Partita No. 3 (*Bourrée*). These three movements are quite interesting for initial experiments because most of the notes are eighth notes,

Table 1
Performers analyzed in the experiments

Ara Malikian	Jacqueline Ross	Rachel Podger
Arthur Grumiaux	James Ehnes	Sergiu Luca
Brian Brooks	Jascha Heifetz	Shlomo Mintz
Christian Tetzlaff	Josef Suk	Sigiswald Kuijken
Garrett Fischbach	Julia Fischer	Susanna Yoko Henkel
George Enescu	Lucy van Dael	Tanya Anisimova
Itzhak Perlman	Mela Tenenbaum	Yehudi Menuhin
Jaap Schroder	Nathan Milstein	

leading us to acquire a model based on many homogeneous segments. The scores of the analyzed pieces are not provided to the system.

Table 1 shows the list of the 23 professional violinists used in our study. From the list, it can be stated that there is a variety of performing styles. Moreover, some of them are from the beginning of the last century while others are modern performers currently active. Violinists like B. Brooks, R. Podger, and C. Tetzlaff play in a Modern Baroque style; whereas violinists like S. Luca, S. Mintz, and J. Ross are well-known for their use of détaché.

We also included two recordings that are clearly different from the others: G. Fischbach and T. Anisimova. G. Fischbach plays with a sustained articulation. T. Anisimova is a cellist and, then, the performance is clearly very different from the others.

4. Trend-based modeling

Because we are using state-of-the-art feature extraction tools and we do not manually process the information we collect, it is not feasible to learn a deep model about each violin performer, i.e. to learn rules mapping specific musical concepts to expressive resources. From one side, we cannot perform a confident musical analysis from the (approximate) score provided by the extraction tools. On the other side, without a high precision in identifying note attacks, we know that the accuracy of expressive features is not guaranteed.

The alternative is to take a more global perspective of a performer by trying to capture their essential (and recurrent) expressive decisions. For instance, playing in a romantic articulation will affect note durations and will produce characteristic energy envelopes. Additionally, the use of *ritardando* produces local changes in note durations and energy strengths.

These expressive decisions are somehow related to the musical analysis each violinist performs of the written score. Nevertheless, we do not have the real score (that one on the music sheet), but only an approximated score derived from the audio. We propose to use a melodic contour segmentation as a way to capture the musical structure of the piece. As most of automatic melody segmentation approaches, we will perform note grouping according to a human perception model. Different levels of abstraction have been proposed for melodic contour analysis. In [7], the advantages and drawbacks of using different abstraction levels have been presented and the Implication/Realization (IR) model [16] has demonstrated the best performance because its capacity of summarizing melodic and rhythmic information. Specifically, the IR model proposes some basic melodic patterns that will be used as an approximation of the local structure of a piece. Moreover, IR model allows a melodic analysis in terms of melodic intervals and relative durations, i.e. precision on score extraction is not critical. Furthermore, in Section 5.3 we show how Narmour's model outperforms the results of a simple contour-based model.

Then, our approach for dealing with the identification of violin performers is based on the acquisition of *trend models* that characterize each particular performer to be identified. Specifically, a trend model characterizes, for a certain audio descriptor, the relationships a given performer is establishing among groups of neighbor musical events. We perform a qualitative analysis of the variations of the audio descriptors. Moreover, as we will describe in the next subsection, we analyze these qualitative variations with a local perspective.

We use two trend models in this paper: energy and duration. The trend model for the energy descriptor relates, qualitatively, the energy variation for a given set of consecutive notes, and it is related to dynamics. On the other hand, the trend model for duration indicates, also qualitatively, how note durations change for note sequences. Duration is related to articulation and timing. Notice that trend models are not trying to characterize the audio descriptors with respect to an expected global behavior.

Given a musical recording of a piece as input, the trend analysis is performed by aggregating the qualitative variations on their small melody segments. Thus, previously to build trend models, input streams are broken down into segments.

Our system has been designed in a modular way with the intention of creating an easy extendable framework. We have three different types of modules in the system: 1) the audio feature extraction and segmentation modules; 2) the trend analysis modules; and 3) the identification modules. Moreover, the system may work in two different modes: training and testing. Modules from (1) and (2) are used in both modes. Modules from (3) are only used in the testing mode. Figure 1 shows a diagram of the system modules. On top, audio files in *.wav* format as input.

At the training stage, the goal of the system is to characterize performers by extracting expressive features and constructing trend models. Next, at the identification stage, the system analyzes the input performance and looks for the most similar previously learned model. The training process is composed of three main steps: 1) the extraction of audio descriptors and the division of a performance in segments; 2) the tagging of segments according to IR patterns; and 3) the computation of probabilistic distributions for each IR pattern and descriptor (trend generation).

4.1. Feature extraction and segmentation

The first step consists on extracting audio features. Our research is not focused on developing new methods for extracting audio features, so that we employ existing techniques. At the moment, we consider fundamental frequency and energy, as these are the main low-level audio features related to melody. These features are then used to identify note boundaries and to generate melodic segments. The current version of our system uses the fundamental frequency estimation algorithm proposed in [2]. This module provides a vector with instantaneous fundamental frequency and energy values computed every 0.01 seconds.

We have developed a post-processing module for determining the possible notes played by the performers. This module first converts fundamental frequencies into quantized pitch values, and then a pitch correction procedure is applied in order to eliminate noise and sudden changes. This correction is made by assigning to each sample the value given by the mode of their neighbors around a certain window of size σ. When some notes are played together we only take into account the one with more energy.

With this process, a smooth vector of pitches is obtained. By knowing on which frames pitches are changing (i.e. consecutive values are different), a note-by-note segmentation of the whole recording is performed. For each note we collect its pitch, duration and energy.

We assume that there might be some errors in this automatic segmentation, given the heterogeneity of recording conditions. Our approach for dealing with this problem consists of using a more abstract

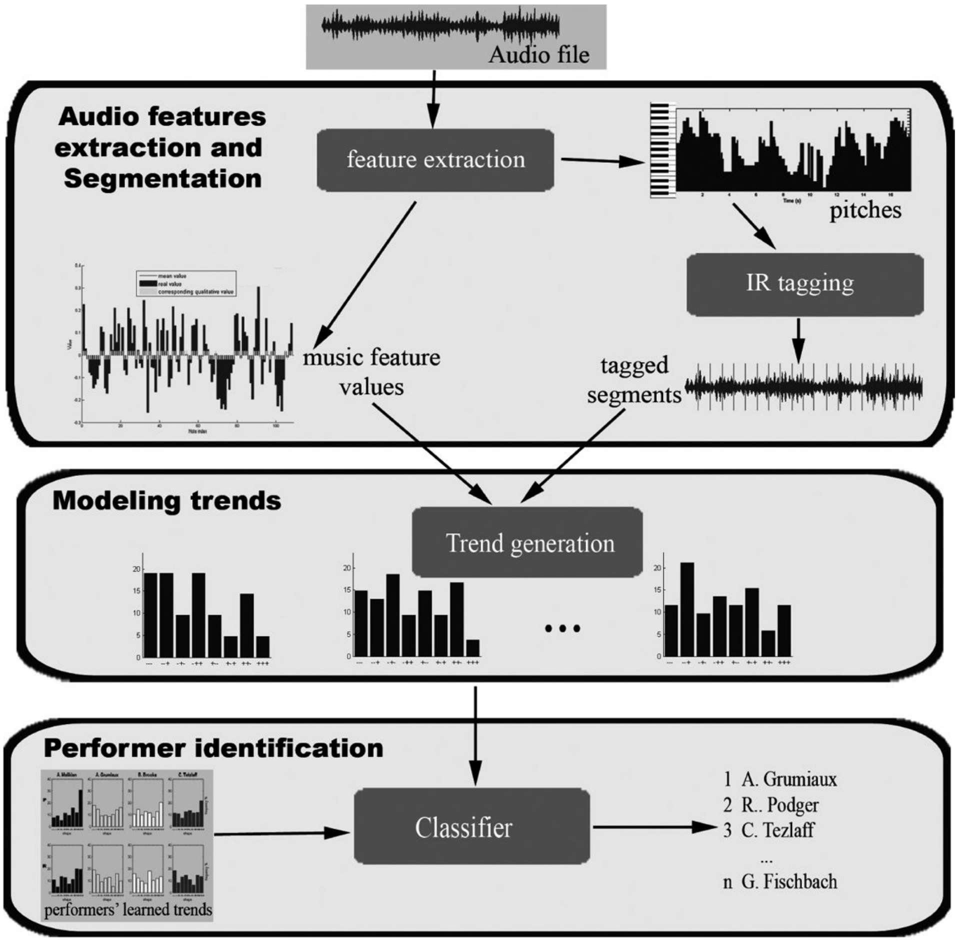


Fig. 1. Architecture of the system.

representation that the real notes, but still close to the melody. That is, instead of focusing on the absolute notes, we are interested in modeling the melodic surface. We use the Implication/Realization model to perform melodic segmentation.

4.1.1. The implication/realization model

The Implication/Realization (IR) Model proposed by Eugene Narmour [15,16] is based on a perceptual and cognitive approach for analyzing the structure of a musical piece. Gratchen et al. [8] showed in MIREX'05 that the IR model is suitable for assessing melodic similarity. Since our goal is to characterize expressive trends, we analyze the way different audio descriptors change in the different IR patterns.

IR model tries to explicitly describe the patterns of expectations generated in the listener with respect to the continuation of the melody. It follows the approach introduced by Meyer [14] that applies the principles of *Gestalt Theory* to melody perception. The model describes both the continuation implied by particular melodic intervals, and the degree to which this expected continuation is actually realized by the following interval.

Gestalt theory states that perceptual elements are grouped together to form a single perceived whole

Table 2
Characterization of the ten IR structures we are able to identify; in the second column, 'S' denotes small, 'L' large, and '0' a unison interval

Structure	Interval sizes	Same direction?	PID satisfied?	PRD satisfied?
P	S S	yes	yes	yes
D	0 0	yes	yes	yes
ID	S S (eq)	no	yes	no
IP	S S	no	yes	no
VP	S L	yes	no	yes
R	L S	no	yes	yes
IR	L S	yes	yes	no
VR	L L	no	no	yes
RP	L L	yes	no	no
RR	S L	no	no	no

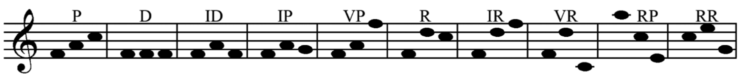


Fig. 2. Ten basic structures of the IR model.

(called 'gestalt'). This grouping follows some principles: *proximity* (two elements are perceived as a whole when they are perceptually close), *similarity* (two elements are perceived as a whole when they have similar perceptual features, e.g. color in visual perception), and *good continuation* (two elements are perceived as a whole if one is a 'natural' continuation of the other).

Narmour claims that similar principles hold for the perception of melodic sequences. In IR, these principles take the form of *implications* and involve two main principles: *registral direction* (PRD) and *intervallic difference* (PID). The PRD principle states that small intervals create the expectation of a following interval in the same registral direction (for instance, a small upward interval generates an expectation of another upward interval), and large intervals create the expectation of a change in the registral direction (for instance, a large upward interval generates an expectation of a downward interval). The PID principle states that a small (five semitones or less) interval creates an expectation of a following similarly-sized interval (plus or minus two semitones), and a large interval (seven semitones or more) creates an expectation of a following smaller interval.

Based on these two principles, melodic patterns can be identified that either satisfy or violate the implication as predicted by the principles. Such patterns are called *structures* and labeled to denote characteristics in terms of registral direction and intervallic difference. The ten basic structures are shown in Fig. 2. For example, the P structure ('Process') is defined as two or more consecutive small intervals (of similar size) satisfying both the registral direction principle and the intervallic difference principle. Similarly, the IP ('Intervallic Process') structure satisfies intervallic difference, but violates registral direction.

Some additional structures are retrospective counterparts of the basic structures. In general, the retrospective variant of a structure has the same registral form and intervallic proportions, but the intervals are smaller or larger. For example, an initial large interval does not give rise to a P structure (rather to an R, IR, or VR, see Fig. 2), but when is followed by another large interval in the same registral direction, the pattern is a pair of similarly sized intervals with the same registral direction and it is identified as a retrospective P structure (denoted as RP). An analogous analysis can be performed with the R structure resulting in a new retrospective pattern (denoted as RR). Table 2 summarizes the characteristics of all these IR structures.

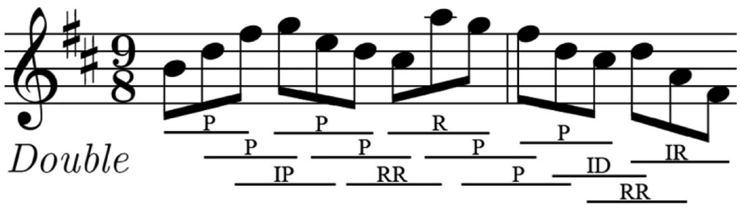


Fig. 3. IR analysis of the beginning of the Sixth Movement (Double) of Partita No. 1.

We have developed an algorithm to automate the annotation of melodies with their corresponding IR analyses. The algorithm implements most of the ‘innate’ processes mentioned before and is able to detect the ten different 3-note patterns described in Table 2: P, D, ID, IP, VP, R, IR, VR, RP and RR. The algorithm uses a sliding window of three notes that moves in steps of one note. For each window, we calculate the size and direction of the two intervals within it, and the IR structure which matches with this 3-note group is selected. In Fig. 3 the IR analysis of the beginning of the Sixth Movement of Partita No. 1 can be found as an example of how we tag each 3-note group.

4.2. Modeling trends

A trend model is represented by a set of discrete frequency distributions for a given audio descriptor (e.g. energy). Each of these frequency distributions represents the way a given IR pattern is played against that certain audio descriptor. Since we are tagging the audio segments with ten different IR patterns, each trend model is represented by ten different frequency distributions.

To generate trend models for a particular performer and audio descriptor, we use the sequences of values extracted from the notes identified in each segment. From these sequences, a qualitative transformation is first performed to the sequences in the following way: each value is compared to the mean value of the fragment and is transformed into a qualitative value where $+$ means ‘the descriptor value is higher than the mean’, and $-$ means ‘the descriptor value is lower than the mean’. Being s the size of the segment and n the number of different qualitative values, there are n^s possible resulting shapes. In the current approach, since we are segmenting the melodies in groups of three notes and using two qualitative values, eight (2^3) different shapes may arise. We note these possibilities as: $---$, $--+$, $-+-$, $-++$, $+--$, $+-+$, $++-$ and $+++$.

Next, a histogram per IR pattern with these eight qualitative shapes is constructed by calculating the percentage of occurrence of each shape. These histograms can be understood as discrete probability distributions. Thus, trend models capture statistical information of how a certain performer tends to play. Combining trend models from different audio descriptors, we improve each performer characterization.

Since our goal is the identification of violin performers, the collection of trend models acquired for each performer is used as the patterns to compare with when a new audio recording is presented to the system.

4.2.1. Current trends

We have generated trend models for both duration and energy descriptors as they are the low-level descriptors more closely related with the melody. Note durations are computed as the number of samples between pitch changes. The mean duration of each note (in samples) is obtained by dividing the total number of samples for the whole fragment by the number of recognized notes in it (obviously taking into account different figures). Because we have computed both the real and the expected (mean) duration

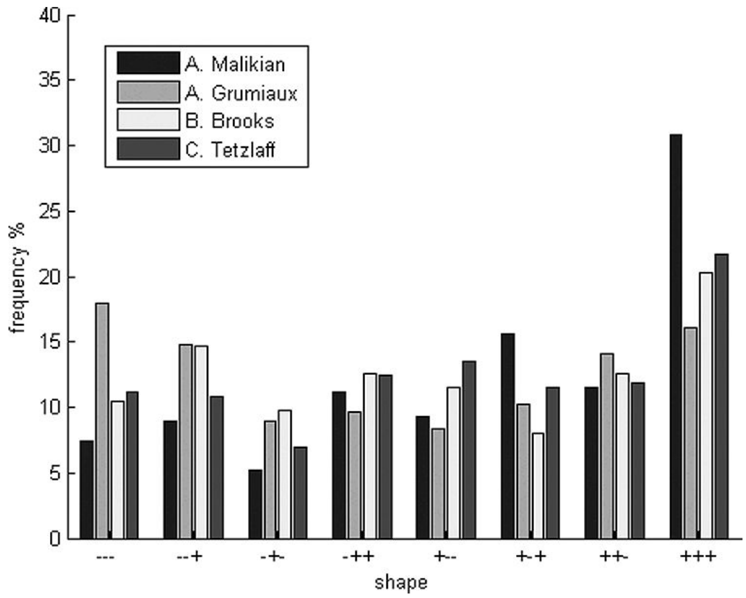


Fig. 4. Frequency distribution of duration deviations for the P pattern in the Sixth movement of Partita No. 1. Only four performers are shown.

for each note we can say whether a note is longer than it should be, or opposite, if it is shorter. So that, the binary vector with qualitative deviations is built and the trend model for the duration descriptor is obtained by matching with the identified IR structures.

Figure 4 shows, for the duration descriptor, the frequency distributions of the eight shapes in P patterns (ascending or descending sequences of small intervals) and for four violin performers (Ara Malikian, Arthur Grumiaux, Brian Brooks, and Christian Tetzlaff).

We observe that the way different professional performers are playing is not equally distributed. For instance, A. Malikian has a higher propensity to extend the durations while an opposite behavior can be observed for A. Grumiaux (see his values for the two left qualitative shapes). It should be noticed that more exact ways of obtaining this measure could be used, as well as taking into account the attack and release times, as other researchers do [19]. This would lead us to a more complex process that we definitively want to avoid by now. Hopefully, we guess that our approach does not need so precise information to identify a given performer.

We have also acquired trend models for the energy descriptor in an analogous way. The energy average for each fragment is calculated and, given the energy of each note, qualitative deviations are computed. Next, from these qualitative values, the trend models are constructed by calculating the frequencies of the eight shapes for each IR pattern.

4.3. Classifying new performances

A nearest neighbor classifier is used to predict the performer of new recordings. Trend models acquired in the training stage, as described in the previous section, are used as class patterns, i.e. each trained performance is considered a different solution class. When a new recording is presented to the system, the feature extraction process is performed and its trend model is created. This trend model is compared with the previously acquired models. The classifier outputs a ranked list of performer candidates where distances determine the order, with 1 being the most likely performer relative to the results of the training phase.

4.3.1. Distance measure

The distance d_{ij} between two trend models i and j (i.e. the distance between two performances), is defined as the weighted sum of distances between the frequency distributions of IR patterns:

$$d_{ij} = \sum_{n \in N} w_{ij}^{n} dist(n_i, n_j) \tag{1}$$

where N is the set of the different IR patterns considered; $dist(n_i, n_j)$ measures the distance between two frequency distributions (see (3) below); and w_{ij}^{n} are the weights assigned to each IR pattern. Weights have been introduced for balancing the importance of the IR patterns with respect to the number of times they appear. Frequent patterns are considered more informative due to the fact that they come from more representative samples. Weights are defined as the mean of cardinalities of respective histograms for a given pattern n:

$$w_{ij}^{n} = (N_i^n + N_j^n)/2 \tag{2}$$

Mean value is used instead of just one of the cardinalities to assure a symmetric distance measure in which w_{ij}^{n} is equal to w_{ji}^{n}. Cardinalities could be different because recognized notes can vary from one performance to another one, even though the score is supposed to be the same.

Finally, distance between two frequency distributions is calculated by measuring the absolute distances between the respective patterns:

$$dist(s, r) = \sum_{k \in K} |s_k - r_k| \tag{3}$$

where s and r are two frequency distributions for the same IR pattern; and K is the set of all possible values they can take (in our case $|K| = 8$).

When both audio descriptors are considered, we simply aggregate the individual corresponding distances.

5. Experiments

The goal of the experiments was to assess the feasibility of our approach. We are aware that additional features could be extracted from the recordings. But combining two basic features such as energy and duration we were interested in measuring the robustness of an identification model that only captures some basic expressive trends.

Experiments consisted in training the system with one movement and, then, testing the acquired trend models with a different movement. We used three different movements in the experiments: the Second and Sixth movements of Partita No. 1 (both a *Double*) and the Fifth movement of Partita No. 3 (*Bourrée*).

Each experiment was performed using three different configurations of trend models. Specifically, the performance of experiments was compared with only using duration-based trend models, only using energy-based trend models, and using both trend models. Finally, we compared the results with only using a contour-based model.

Table 3
Number of different IR structures identified in each of the three movements studied. The values are the mean of the 23 recordings

	P	D	ID	IP	VP	R	IR	VR	RP	RR
P.1 No.2	287	0	187	222	42	89	56	0	113	100
P.1 No.6	257	0	156	157	26	56	28	0	66	64
P.3 No.5	256	0	125	145	18	44	23	0	52	55

Table 4
Success rate (%) in all experiments taking into account three different ranking positions proposed for the correct performer: 1st, 3rd, and 10th

	set-1			**set-2**			**set-3**		
	1st	3rd	10th	1st	3rd	10th	1st	3rd	10th
duration	34.8	60.9	95.7	21.7	43.5	91.3	10.5	26.3	68.4
energy	21.7	69.6	91.3	30.4	47.8	91.3	15.8	31.6	73.7
both	52.2	65.2	91.3	34.8	47.8	95.7	15.8	26.3	68.4

5.1. *Acquiring trend models*

The generation of trend models starts with the segmentation of the recordings at two levels: note events and IR patterns. At the low level, the pitch estimation algorithm segments a recording with a sequence of possible notes. Comparing the number of notes automatically detected with the real scores, the pitch estimator is generating more segments than the notes in the score (sometimes doubling the number of notes). This result is not surprising because we are not explicitly dealing with expressive resources like vibratos or portamentos. Thus, some IR patterns will contain these expressive resources. Nevertheless, because we are not explicitly working with notes, these expressive patterns will also contribute to the characterization of performing styles. For instance, a portamento is usually captured as a sequence of notes where one (or some) have a short duration.

Table 3 details the number (in mean) of IR segments detected for each movement. Observe that there are two IR structures (D and VR) that are not present in the recordings. The most recurrent IR structure is the P pattern followed by the ID pattern. This result is coherent with the fact that the three movements contain a lot of arpeggios.

Finally, trend models of performers are built from the way these 710 to 1.000 IR structures are performed in terms of energy and duration.

In order to better understand what the different trend models are capturing, we obtained the distances d_{ij} between all of them (see Eq. (1)) and applied a hierarchical clustering algorithm (using a complete linkage method). Figure 5 shows the dendrogram representation of the hierarchical clustering for the Sixth movement of Partita No. 1. It is interesting to remark that some performing styles are captured by these trend models. For instance, the most dissimilar recordings (G. Fischbach and T. Anisimova) are clearly far from the rest; violinists playing in a Modern Baroque style (B. Brooks, R. Podger, and C. Tetzlaff) are clustered together; violinists using détaché (S. Luca, S. Mintz, and J. Ross) also appear close to each other; and the usage of expressive resources such as portamento, vibrato, or ritardando presents a relationship with the clustering result.

5.2. *Results*

We performed two different types of experiments. The first experiment was focused on assessing the performance of the system by using two movements from the same piece. Specifically, we used the

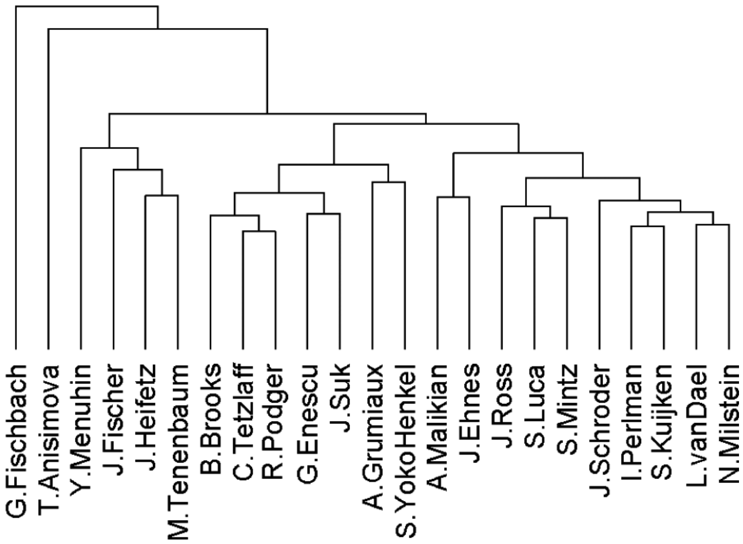


Fig. 5. Hierarchical clustering for the Sixth movement of Partita No. 1.

Second and the Sixth movements of Partita No. 1. In the following, we will call **set-1** the experiment where 23 instances of the second movement were used for training and 23 from the sixth for testing. Analogously, we will call **set-2** the experiment where the sixth movement was used for training and the second for testing.

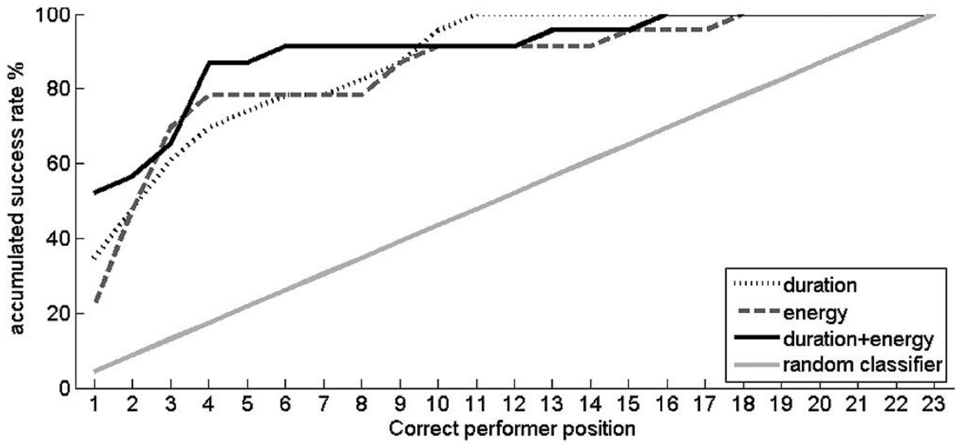
The second type of experiments was focused on assessing the performance of the system by using two movements from different pieces. Specifically, we used the second movement of Partita No. 1 for training and the fifth movement of Partita No. 3 for testing. We will refer to this test as **set-3**.

For each input recording, the system result is a ranked list of performers sorted from the most similar to the furthest one to the input. The highest accuracy is achieved when the correct performed is the first of the list. Otherwise, a last position for the correct performer represents the worst accuracy.

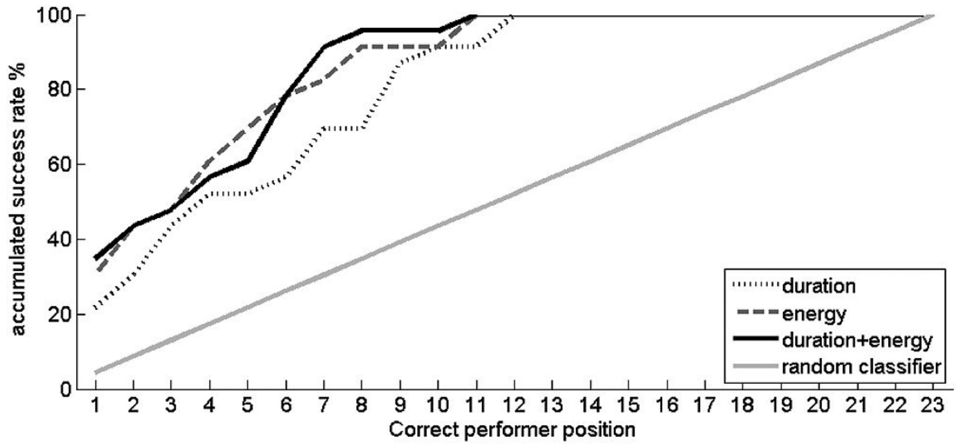
A complete view of the results is shown in Fig. 6 and summarized in Table 4. Figure 6 shows the percentage of input recordings identified at each position. It provides a picture of the system accuracy using the length of the proposed ranking as a threshold. Table 4 summarizes the performance of the system for the three experimental sets and the three trend models. The three columns of each experiment show, respectively, the percentage of performers identified in the first position, at least in the third position, and at least in the tenth position.

Regarding the experiments with movements from the same Partita (experiments **set-1** and **set-2**), the correct performer was mostly identified in the first half of the list, i.e. at most in the 12th position. The correct performer is predicted, in the worst case, 20% of times as the first candidate, clearly outperforming the random classifier (whose success rate is 4.3%). Additionally, using the four top candidates the accuracy reaches the 50% of success.

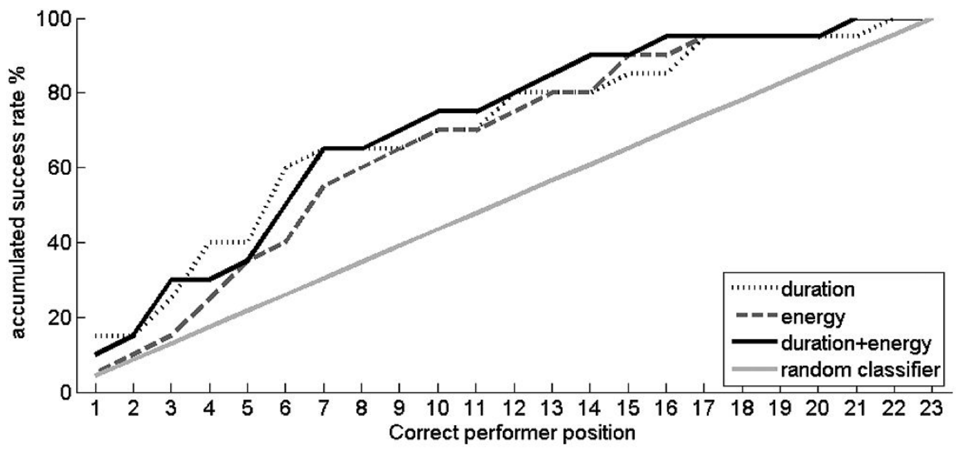
Regarding experiment **set-3**, the most difficult scenario, the 90% of identification accuracy was overcame at position 15. The 50% of success for the trend models based on only one feature (duration or energy) is achieved by selecting seven performers. Combining both features the accuracy overcomes 60%. The results are promising, especially comparing with a random classification where the success rate is clearly outperformed.



(a) Partita No. 1 Mov. 2 vs. Partita No. 1 Mov. 6 (set-1)



(b) Partita No. 1 Mov. 6 vs. Partita No. 1 Mov. 2 (set-2)



(c) Partita No. 1 Mov. 2 vs. Partita No. 3 Mov. 5 (set-3)

Fig. 6. Accumulated success rate by position of the correct performer in all the performed experiments.

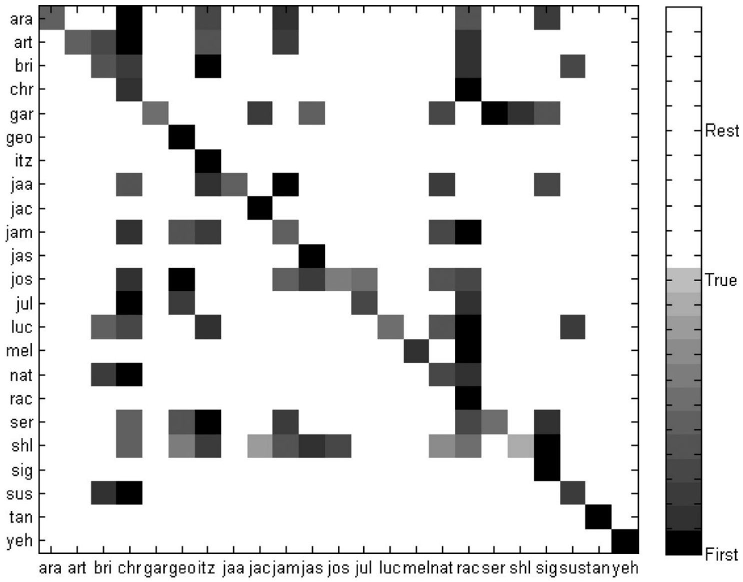


Fig. 7. Classifier output in matrix form for **set-1** where both (duration and energy) trend models were used.

Figure 7 presents a matrix that summarizes the classifier output for **set-1** using both duration and energy trend models. The figure details the information given by the ‘duration+energy’ curve in Fig. 6a. Specifically, it shows, for each input recording (row), the sorted list of predicted performers as squares. Ranking values are mapped onto a gray scale. The black color indicates the first performer proposed and the gray degradation is used to draw all the performers predicted until the correct one. Notice that the success in the first position means a black square in the diagonal. The matrix is not supposed to be symmetric and each column can have the same color several times because a predicted performer can occur in the same position for several inputs. For instance, we can see that Garret Fischbach’s performance (`gar`) for Sixth Movement is very different from the rest of performers’ Second Movement performances: all values correspond to distance positions. On the other hand, Christian Tetzlaff’s (`chr`) and Rachel Podger’s (`rac`) performances are quite similar to most of Second Movement performances since there are many squares in their columns.

Finally, Fig. 8 shows in which position the correct performer is ranked for each performer in the test set. This Figure complements the former two ones. The results came from **set-1** using both trend models (‘duration+energy’ curve in Fig. 6a). Twelve right identifications were achieved at first position (52% of success rate). The rest was correctly identified in positions 2 to 4 except three performers. Nathan Milstein was identified at position 6. Finally, Sergiu Luca and Shlomo Mintz were not clearly identified. After a detailed analysis of the distances among all performers, we observed that these two musicians are not clearly distinguished when using a nearest neighbor classifier. Their performances, with respect to duration and energy, are close to multiple other performers.

5.3. Analyzing a contour-based approach

In order to validate the IR approach, we performed experiments for assessing the performance of a contour-based segmentation. The contour segmentation provides nine classes regarding the direction of the two existing intervals within each 3-note group: two ascending intervals, two descending, one up

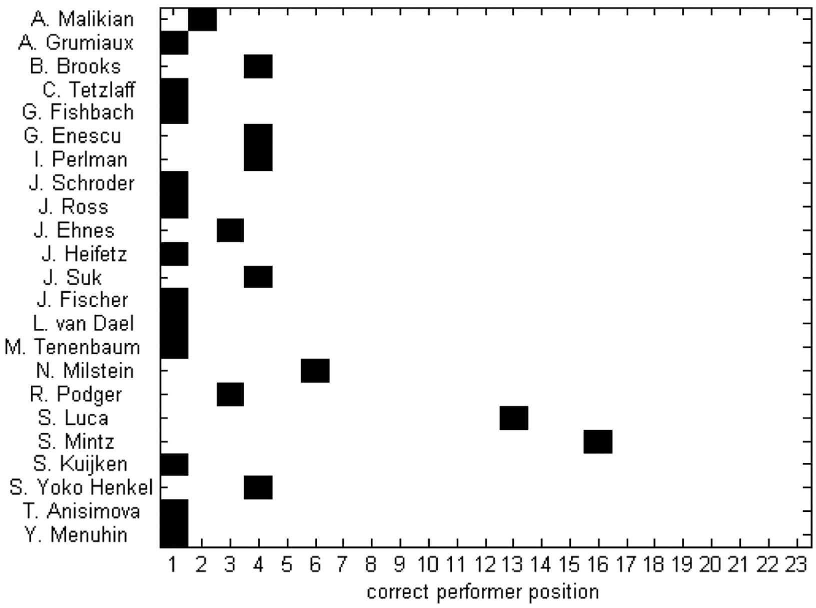


Fig. 8. Correct performer position for each performance in **set-1**. Both trend models are used.

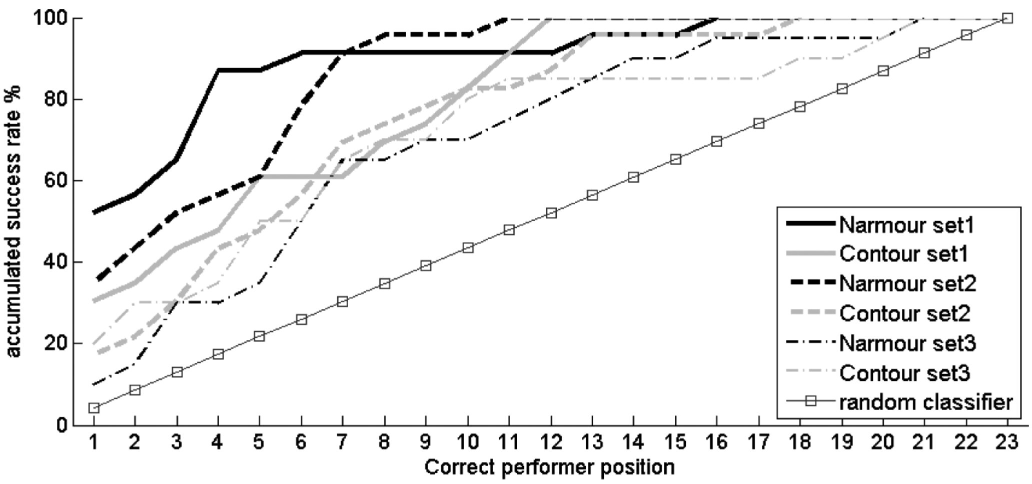


Fig. 9. Comparison of results by using Narmour segmentation and simple contour segmentation for **set-1**, **set-2** and **set-3**. Both trend models (duration and energy) were used.

and one down, two unisons, and so on. Figure 9 provides a comparison between the classification results using IR trends versus contour trends. The contour-based segmentation also outperforms the random classifier. Nevertheless, the IR-based models present better performance than the contour-based models. The results are not surprising because IR models are constructed with a finer interval analysis. Moreover, we have to stress that the contour-based results profit from the rhythmic regularity (most of the notes are eights), i.e. in the general case results will be worse.

6. Conclusions

This work focuses on the task of identifying violinists from their playing style by building trend-based models that capture expressive tendencies. Trend models are acquired by using state-of-the-art audio feature extraction tools and automatically segmenting the obtained melodies using IR patterns. Performers were characterized by a set of frequency distributions, capturing their personal style with respect to a collection of melodic patterns (IR patterns). We have shown that, without a great analysis accuracy, our proposal is quite robust.

The experiments were concentrated on identifying violinists and using note durations and energies as descriptors. We tested the system with 23 different professional performers and different recordings. Results obtained show that the proposed model is capable of learning performance patterns that are useful for distinguishing performers. The results clearly outperform a random classifier and, probably, it would be quite hard for human listeners to achieve such recognition rates. In order to assure the robustness of the system, other sets of works should be used for learning and testing.

Our current experiments have been constrained to monophonic audio. We would like to extend the method in order to deal with polyphonic recordings in an appropriate way.

We have presented a qualitative analysis using only two qualitative values. We want to keep our model at this qualitative level but we plan to extend the model with the use of fuzzy sets. This improvement will allow us to use the capabilities of fuzzy theory for a better assessment in the similarity measure.

Combining information from different music features has been demonstrated to improve results. We are currently working on increasing the number of descriptors. Since the predictability of a given descriptor varies depending on the performers, we are also interested in discovering relations among the descriptors. Finally, the use of hierarchical classifiers or ensemble methods is a possible way to improve the identification accuracy.

Acknowledgements

The authors want to thank Agustin Martorell for his valuable comments regarding the musicologist meaning of the expressive resources used by the violinists in the recordings. Miguel Molina-Solana is supported by the Spanish Ministry of Education and Science (project TIN2006-15041-C04-01 and FPU grant AP2007-02119). Josep Lluís Arcos is supported by the Spanish project MID-CBR (TIN2006-15140-C03-01), EU-FEDER funds, and by the Catalan Government (2005-SGR-00093). Emilia Gomez is supported by EU-eContent-Plus project VARIAZIONI[1] (ECP-2005-CULT-038264).

References

[1] Josep Lluís Arcos, Ramon López de Mántaras and Xavier Serra, Saxex: a case-based reasoning system for generating expressive musical performances, *Journal of New Music Research* **27**(3) (1998), 194–210.

[2] Arturo Camacho, *SWIPE: A Sawtooth Waveform Inspired Pitch Estimator for Speech and Music*, PhD thesis, University of Florida, USA, 2007.

[3] Ramon Lopez de Mantaras and Josep Lluis Arcos, AI and music: from composition to expressive performance, *AI Magazine* **23**(3) (2002), 43–57.

[1] http://www.variazioniproject.org

[4] Matthew J. Dovey, Analysis of Rachmaninoff's piano performances using inductive logic programming, In *Proc. of the 8th European Conference on Machine Learning* (*ECML95*), pages 279–282, London, UK, 1995. Springer-Verlag.

[5] Werner Goebl, Analysis of piano performance: Towards a common performance standard? In *Proc. of the Society for Music Perception and Cognition Conference* (*SMPC99*), Evanston, USA, 1999.

[6] F. Gouyon, P. Herrera, E. Gómez, P. Cano, J. Bonada, A. Loscos, X. Amatriain and X. Serra, *Content Processing of Music Audio Signals*, (Chapter 3), Logos Verlag Berlin GmbH, Berlin, 2008, pp. 83–160.

[7] Maarten Grachten, Josep Lluis Arcos and Ramon Lopez de Mantaras, Melodic similarity: Looking for a good abstraction level, pages 210–215, 2004.

[8] Maarten Grachten, Josep Lluis Arcos and Ramon Lopez de Mantaras, Melody retrieval using the Implication/Realization Model. In *6th International Conference On Music Information Retrieval* (*ISMIR 2005*). (*First prize of the MIREX Symbolic Melodic Similarity Contest*), 2005.

[9] Maarten Grachten, Josep Lluis Arcos and Ramon Lopez de Mantaras. A case based approach to expressivity-aware tempo transformation, *Machine Learning* **65**(2-3) (2006), 411–437.

[10] Ju-Lee Hong, Investigating expressive timing and dynamics in recorded cello, *Psychology of Music* **31**(3) (2003), 340–352.

[11] Patrik N. Juslin, Anders Friberg and Roberto Bresin, Toward a computational model of expression in performance: The GERM model, *Musicae Scientiae*, special issue: 63–122, 2002.

[12] Patrik N. Juslin and John A. Sloboda, *Music and Emotion: Theory and Research*, Oxford University Press, 2001.

[13] Joel Lester, *Bach's Works for Solo Violin: Style, Structure, Performance*, Oxford University Press, 1999.

[14] Leonard Meyer, *Emotion and Meaning in Music*, Chicago, IL: Univ. Chicago Press, 1956.

[15] Eugene Narmour, *The Analysis and Cognition of Basic Melodic Structures: The Implication Realization Model*, Chicago, IL: Univ. Chicago Press, 1990.

[16] Eugene Narmour, *The Analysis and Cognition of Melodic Complexity: The Implication Realization Model*, Chicago, IL: Univ. Chicago Press, 1992.

[17] Montserrat Puiggros, Comparative analysis of expressivity in recorded violin performances. Study of the sonatas and partitas for solo violin by J. S. Bach. Master's thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2007.

[18] Montserrat Puiggros, Emilia Gómez, Rafael Ramirez, Xavier Serra and Roberto Bresin, Automatic characterization of ornamentation from bassoon recordings for expressive synthesis, In *Proc. of 9th International Conference on Music Perception and Cognition* (*ICMPC 2006*), Bologna, Italy, 2006.

[19] Rafael Ramirez, Esteban Maestre, Antonio Pertusa, Emilia Gómez and Xavier Serra, Performance-based interpreter identification in saxophone audio recordings, *IEEE Trans. on Circuits and Systems for Video Technology* **17**(3) (2007), 356–364.

[20] Rafael Ramirez, Alfonso Perez, Stefan Kersten and Esteban Maestre, Performer identification in celtic violin recordings. In *Proc. of 9th International Conference on Music Information Retrieval* (*ISMIR 2008*), pages 483–488, Philadelphia, USA, 2008.

[21] Craig Sapp, Comparative analysis of multiple musical performances, In *Proc. of 8th International Conference on Music Information Retrieval* (*ISMIR 2007*), pages 497–500, Vienna, Austria, 2007.

[22] Craig Saunders, David Hardoon, John Shawe-Taylor and Gerhard Widmer, Using string kernels to identify famous performers from their playing style, *Intelligent Data Analysis* **12**(4) (2008), 425–440.

[23] Xavier Serra, Gerhard Widmer and Marc Leman, *A roadmap for sound and music computing*, The S2S2 Consortium, 2007.

[24] Efstathios Stamatatos and Gerhard Widmer, Automatic identification of music performers with learning ensembles, *Artificial Intelligence* **165**(1) (2005), 37–56.

[25] Neil P. Todd, The dynamics of dynamics: A model of musical expression, *Journal of the Acoustical Society of America* **91**(6) (1992), 3540–3550.

[26] Vesa Valimaki, Rudolf Rabenstein, Davide Rocchesso, Xavier Serra and Julius O. Smith, Signal processing for sound synthesis: Computer-generated sounds and music for all, *IEEE Signal Processing Magazine* **24**(2) (2007), 8–10.

[27] Gerhard Widmer, Simon Dixon, Werner Goebl, Elias Pampalk and Asmir Tobudic, In search of the Horowitz factor, *AI Magazine* **24**(3) (2003), 111–130.