Concept Drift Detection in Discrete Streaming Data Using Probabilistic Graphical Models

Rafael Cabañas[1], Andrés Cano[1], Manuel Gómez-Olmedo[1], Andrés R. Masegosa[2], and Serafín Moral[1]

[1] Department of Computer Science and Artificial Intelligence, CITIC, University of Granada, Spain
{rcabanas,acu,mgomez,smc}@decsai.ugr.es

[2] Department of Statistics and Applied Mathematics, University of Almeria, Spain
andresmasegosa@ual.es

Abstract. A common problem in mining data streams is that the distribution of the data might change over time. This situation, which is known as concept drift, should be detected for ensuring the accuracy of the models. In this paper we propose a method for concept drift detection in discrete streaming data using probabilistic graphical models. In particular, our approach is based on the use of conditional linear Gaussian Bayesian networks with latent variables. We demonstrate and analyse the proposed model using synthetic and real data.

Keywords: concept drift, data stream, Bayesian networks, latent variables, conditional linear Gaussian.

1 Introduction

In recent years, the field of mining data streams has received an increasing attention as large amounts of data are continuously being generated. For example, at financial sector, at social networks, etc. In general, streaming data is open ended, i.e., continuously grows in size. An important aspect of data streams is that the domain being modelled is often *non-stationary*. In other words, the distribution governing the data changes over time. This situation is known as *concept drift* [13, 15, 5] and if not carefully taken into account, the result can be a failure to capture and interpret intrinsic properties of the data.

Here we propose a method for concept drift detection in discrete streaming data. For that, we use the conditional linear Gaussian (CLG) model, which is a kind of Bayesian network with continuous and discrete nodes. Our model is based on the one proposed by Borchani et al. [1, 4], an approach using the CLG model and latent variables that was applied to continuous variables. Basically, we propose transforming the discrete data into continuous and then applying a similar approach with latent variables.

The paper is organized as follows. Section 2 introduces some basic concepts. Section 3 details our approach for concept drift detection. The empirical analysis is presented in Section 4. Finally, the conclusions are given in Section 5.

2 Preliminaries

2.1 Data streams with concept drift

Let us first introduce the basic notation. We use upper-case letters for random variables and lower-case for their possible values. For example, x is a value of a given variable X. The set of all possible values that a variable X can take is called domain and denoted Ω_X. For the sets of variables and their assignments we use boldface letters, e.g, the set of variables $\boldsymbol{X}$ takes the values in $\boldsymbol{x}$. In general, a data stream is observed at time-points $t_1, t_2, \ldots$ where $t_j < t_{j+1}$ for all j. We have at each time point t a collection of instances (a.k.a window or batch) denoted $\boldsymbol{x}^t := \{\boldsymbol{x}^t[1], \boldsymbol{x}^t[2], \ldots, \boldsymbol{x}^t[N_t]\}$.

Data streams are usually non-stationary, which implies changes in the statistical properties of the data stream over time. This is known as *concept drift* [13, 15, 5]. More formally, if concept drift is present in a data stream defined over $\boldsymbol{X}$, it holds that $P_{t_j}(\boldsymbol{x}) \neq P_{t_k}(\boldsymbol{x})$ where $P_t(\boldsymbol{x})$ denotes the joint distribution over $\boldsymbol{x}$ at time t, and where t_j, and t_k are 2 different time-points. In what follows we shall consider that concept drift only happens across time and not within a set of instances belonging to the same time-point. For a better understanding of the idea of concept drift, let us consider Fig. 1.a which depicts the evolution of a continuous variable $Y \sim \mathcal{N}(\mu_Y, \sigma_Y)$. In particular, it shown the empirical mean μ_Y calculated with each batch of instances. Notice that there are substantial shifts at batches 2, 4, 9. Similarly, Fig. 1.b shows the evolution of a distribution defined over the discrete variable X whose domain is $\Omega_X = \{x_1, x_2, x_3\}$. In batch 4, there are drastic variations in the estimated values for $P(X = x_1)$ and $P(X = x_2)$ but not for $P(X = x_3)$. When dealing with discrete domains, the changes in probability distributions may affect only to a subset of the domain. On the other hand, at batch 11 the changes affect to the whole domain.

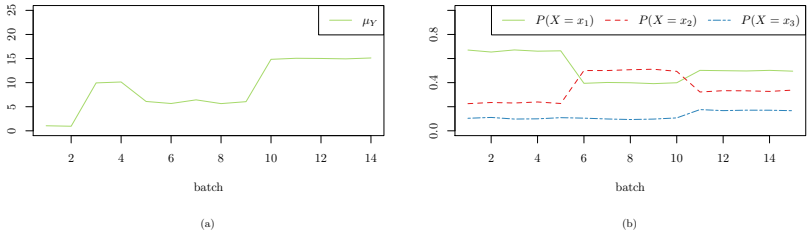

Fig. 1. Concept drift example in (a) continuous and in (b) discrete domains.

2.2 Bayesian networks

Our approach for concept drift detection is based on *Bayesian networks* (BNs) [12, 6], which are a class of PGMs representing a joint probability distribution over a finite set of random variables. The nodes represent the variables in the

problem being modelled, and the links represent the (conditional) dependencies and independencies among the variables.

Definition 1 (Bayesian network). *A Bayesian network (BN) is a tuple* $\langle \boldsymbol{X}, \mathbf{P}, \mathcal{G} \rangle$ *where:* $\boldsymbol{X}$ *is a set of discrete random variables;* $\mathcal{G}$ *is a DAG where each node represents a variable in* $\boldsymbol{X}$*;* $\mathbf{P}$ *is a set of conditional probability distributions, containing one distribution* $P(X|pa(X))$ *for each* $X \in \boldsymbol{X}$ *where* $pa(X)$ *is the set of parents of* X *according to* $\mathcal{G}$*.*

Traditionally, BNs have been defined for discrete domains, which implies several restrictions in problems that can be modelled. For that reason, we will consider conditional linear Gaussian (CLG) BNs [7, 8], which are an extension of BNs allowing discrete and continuous variables. The conditional probability distributions of continuous variables are specified as CLG distributions and discrete variables can only have discrete parents. The conditional distribution of each discrete variable $X_D \in \boldsymbol{X}$ given its parents is a multinomial. On the other hand, the conditional distribution of each continuous variable $Z \in \boldsymbol{X}$ with discrete parents $\boldsymbol{X}_D \subseteq \boldsymbol{X}$ and continuous parents $\boldsymbol{X}_C \subseteq \boldsymbol{X}$, is given by

$$p(z|\boldsymbol{X}_D = \boldsymbol{x}_D, \boldsymbol{X}_C = \boldsymbol{x}_C) = \mathcal{N}(z; \alpha(\boldsymbol{x}_D) + \beta(\boldsymbol{x}_D)^\intercal \boldsymbol{x}_C, \sigma(\boldsymbol{x}_D)), \tag{1}$$

for all $\boldsymbol{x}_D \in \Omega_{\boldsymbol{X}_D}$ and $\boldsymbol{x}_C \in \Omega_{\boldsymbol{X}_C}$,where α and β are the coefficients of a linear regression model of Z given its continuous parents. Fig. 2 depicts two examples. Note that the BN on the right contains a *latent (i.e., hidden)* variable [11] depicted as a white node. A variable of this kind cannot be directly observed but we may introduce it to make our model more powerful. The rest of the variables are called observed and will be represented with nodes in grey.

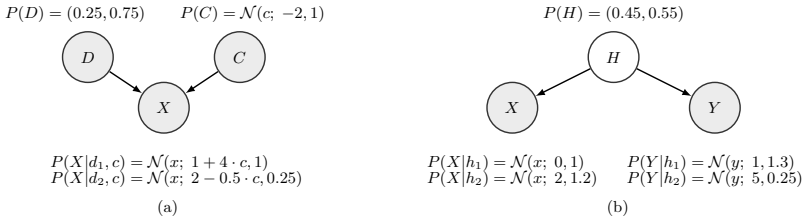

Fig. 2. Example of 2 conditional linear Gaussian BNs.

The BN in Fig. 2.b defines a multivariate Gaussian mixture [11, page 339], which is a latent variable model typically used for unsupervised clustering. Here, the latent variable H represents the clusters or groups while variables X and Y are the data attributes. Note that, if H were continuous, its interpretation would be a variable that summarizes all the data attributes. This is the key idea in the model here proposed for concept drift detection.

In the task of learning BNs from streaming data, the following problem appears. It is not possible to learn the model with the whole data, which might not have been generated yet or it cannot be stored in the memory due to its size.

For that reason, some scalable methods for learning BNs from data streams have been developed in the last years, allowing to efficiently update the model when new data is available. Some of these methods are *variational message passing (VMP)* algorithm [16], a parallel implementation called *d-VMP* [9, 10] and the *streaming variational Bayes* method [2].

3 Concept drift detection with latent variables

Our model for concept drift detection is based on the one proposed by Borchani et al. [1, 4]. This previous approach is defined in the context of classification where $\boldsymbol{X} = \{X1, X2, \ldots, Xn\}$ is the set of (continuous) predictive attributes and a discrete class C. Thus, we at each time point t a collection $(\boldsymbol{x}^t[i], c^t[i])$ for $i = 1$ to N_t. Fig. 3 shows the BN with plate (a.k.a. plateau) notation [3] proposed by Borchani et al. [1] for modelling concept drift.

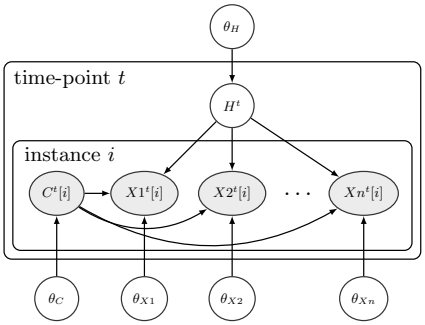

Fig. 3. Model for concept drift detection proposed by Borchani et al. [1].

In the previous model, the observed variables are the predictive attributes and the class variables. A continuous latent variable H^t is set as a parent of all the predictive attributes. In addition, the parameters are represented with each of the nodes labelled with θ. For determining the presence of concept drift, we must estimate the posterior distributions of the H^t-variable at each time-point. A variation on its the expected value implies that $P(X1, X2, \ldots, Xn)$ drifts.

In the present paper we propose a method for detecting concept drift in data streams with discrete variables. In CLG models, discrete variables cannot have continuous parents. As a consequence, the model in Fig. 3 cannot be directly applied to discrete domains. We propose transforming the data to continuous data before learning the model. For that, we apply Algorithm 1 for transforming each batch of discrete data $\boldsymbol{x}$, into equivalent numerical data $\boldsymbol{x}'$ that will be considered as continuous.

For example, let us consider a data stream defined over $\boldsymbol{X} = \{X, Y\}$ with $\Omega_X = \{x_1, x_2, x_3\}$ and $\Omega_Y = \{y_1, y_2, y_3, y_4\}$. Then, the resulting data will be defined over the set of continuous variables $\boldsymbol{X}' = \{X1, X2, X3, Y1, Y2, Y3, Y4\}$. Table 1 shows an example of this transformation.

Algorithm 1 pre-processing

input : $\boldsymbol{x} := \{\boldsymbol{x}[1], \boldsymbol{x}[2], \ldots, \boldsymbol{x}[N]\}$ (batch of discrete instances over $\boldsymbol{X}$)
output : $\boldsymbol{x}' := \{\boldsymbol{x}'[1], \boldsymbol{x}'[2], \ldots, \boldsymbol{x}'[N]\}$ (batch of continuous instances $\boldsymbol{X}'$)

for $i \leftarrow 1$ **to** N **do**
  $\boldsymbol{x}'[i] \leftarrow \emptyset$
  for each $X \in \boldsymbol{X}$ **do**
    for $j \leftarrow 1$ **to** $|\Omega_X|$ **do**
      Let $x[i]$ the value of X in the instance $\boldsymbol{x}[i]$
      Let x_j the j^{th} state in Ω_X
      if $x[i] = x_j$ **then**
        $\boldsymbol{x}'[i] \leftarrow \boldsymbol{x}'[i] \cup \{1\}$
      else
        $\boldsymbol{x}'[i] \leftarrow \boldsymbol{x}'[i] \cup \{0\}$
      end if
    end for
  end for
end for
return $\boldsymbol{x}'$

$\boldsymbol{x}[i]$	$\boldsymbol{x}'[i]$
$\{X = x_1,\ Y = y_1\}$	$\{X1 = 1,\ X2 = 0,\ X3 = 0,\ Y1 = 1,\ Y2 = 0,\ Y3 = 0,\ Y4 = 0\}$
$\{X = x_3,\ Y = y_1\}$	$\{X1 = 0,\ X2 = 0,\ X3 = 1,\ Y1 = 1,\ Y2 = 0,\ Y3 = 0,\ Y4 = 0\}$
$\{X = x_2,\ Y = y_4\}$	$\{X1 = 0,\ X2 = 1,\ X3 = 0,\ Y1 = 0,\ Y2 = 0,\ Y3 = 0,\ Y4 = 1\}$
$\{X = x_1,\ Y = y_3\}$	$\{X1 = 1,\ X2 = 0,\ X3 = 0,\ Y1 = 0,\ Y2 = 0,\ Y3 = 1,\ Y4 = 0\}$

Table 1. Example of output of Algorithm 1

Once that the data is transformed, we can consider a similar model with latent variables. Fig. 4 depicts the proposed BN for detecting concept drift in a data stream with two discrete variables X and Y. There are however some differences w.r.t. to the one by Borchani et al. [1]. First, the model here proposed may contain multiple H^t-variables. Moreover, this latent variables are not parent of all the observed variables. This allows us to detect concept drift in different subsets of the domains in discrete variables. That is, when analysing a data stream we might be interested only in some states in the domains while the rest can be ignored. For example, in Fig. 4, variations in $H2^t$ implies changes in the probability values $P(X = x_2)$, $P(Y = y_1)$ or $P(Y = y_n)$. The number of latent variables and their outgoing arcs are defined by the user.

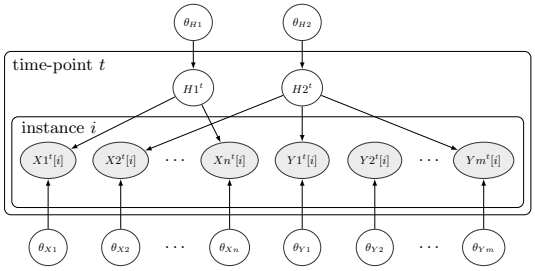

Fig. 4. Model for concept drift detection in a data stream with two discrete variables.

In this model none of the nodes corresponds to a class variable: it can be either dropped from the model or treated as the rest of the discrete variables. Thus, our model can be applied to a higher number of problems, not only to classification tasks. In addition, this also allows to simplify the model and hence make the processing more efficient.

4 Empirical validation

Herein we empirically test our approach. We consider a synthetic data stream and another one including information about intrusion detection in a web server. In both cases, we show the particular models and the evolution of the latent variables modelling concept drift. The experimentation was done using the AMIDST Toolbox[3] and all the material for its replication is available at GitHub[4].

4.1 Synthetic data stream

The justification for testing our approach with a randomly generated data stream is that we can control the underlying distributions of the data. At certain time-points, the probability distributions used for sampling are changed in order to simulate the presence of concept drift. Here, we consider two discrete variables X and Y with 3 and 4 states respectively. The data set contains a total of 12000 instances sampled from the distributions shown in Table 2. We consider that each time-step contains 1000 instances (i.e., size of the window or batch).

	time-step t											
	1	2	3	4	5	6	7	8	9	10	11	12
$P(X = x_1)$	0.2	0.2	**0.6**	0.6	**0.8**	0.8	**0.2**	0.2	0.2	0.2	0.2	0.2
$P(X = x_2)$	0.2	0.2	0.2	0.2	**0.0**	0.0	0.0	0.0	**0.5**	0.5	0.5	0.5
$P(X = x_3)$	0.6	0.6	**0.2**	0.2	0.2	0.2	**0.8**	0.8	**0.3**	0.3	0.3	0.3
$P(Y = y_1)$	0.4	0.4	0.4	0.4	0.4	0.4	**0.2**	0.2	**0.0**	0.0	0.0	0.0
$P(Y = y_2)$	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	**0.6**	0.6	0.6	0.6
$P(Y = y_3)$	0.1	0.1	0.1	0.1	0.1	0.1	**0.3**	0.3	0.3	0.3	0.3	0.3
$P(Y = y_4)$	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1

Table 2. Multinomial distributions for sampling the synthetic data. Values shown in bold indicates the variations in the probabilities.

As explained in the previous section, we have to apply Algorithm 1 to each batch in the data stream. The result is a stream defined over the set of continuous variables $\{X1, X2, X3, Y1, Y2, Y3, Y4\}$. Fig. 5 shows the BN for detection of concept drift in this data stream. Note that variations in variable $H1^t$ implies changes in $P(X = x_1)$ or $P(X = x_3)$. On the other hand, $H2^t$ detects the variations in $P(X = x_2)$, $P(Y = y_2)$ or $P(Y = y_3)$.

[3] `http://www.amidsttoolbox.com`
[4] `https://github.com/PGMLabSpain/2017-CDdiscrete-Code`

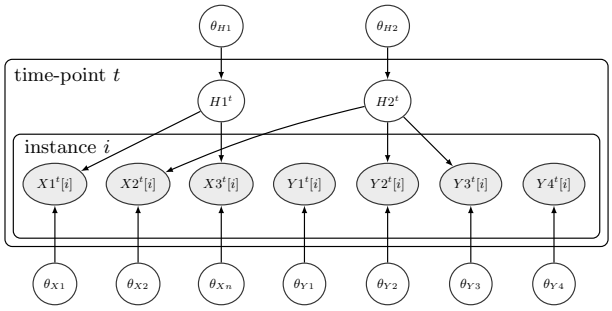

Fig. 5. Proposed model for concept drift detection in the synthetic data stream generated using the distributions given in Table 2.

The evolution of the expected values of the hidden variables $H1^t$ and $H2^t$ are shown in Fig. 6. As can be observed, the changes in the probabilities of interest are proportionally reflected in variations of the latent variables. For example, the most significant variations in $H1^t$ correspond to the large variations of $P(X = x_1)$ or $P(X = x_3)$ at time points 3,7 and 9. On the other hand, at time point 5, $P(X = x_1)$ barely changes while $P(X = x_3)$ does not vary. This implies a very small variation in $H1^t$. If we analyse the evolution of $H2^t$, the single significant variation occurs at time point 9, which corresponds to a large variation in $P(X = x_2)$. In the probabilities of $P(Y = y_2)$ and $P(Y = y_3)$ there are not drastic variations.

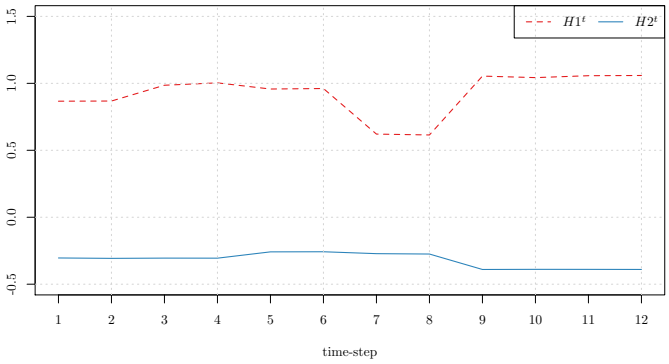

Fig. 6. Results for the synthetic data stream.

4.2 Intrusion detection data stream

Due to its simplicity, the previous data stream is useful for analysing the behaviour of our approach. However, now we aim to test it with large real-world

data. Herein we use a modified version of the intrusion detection data from KDD Cup 1999 competition[5][14]. Each instance corresponds to a connection to a web server. It contains 494021 instances with 42 variables. Yet, we only consider the discrete variables $V1$, $V2$ and $V3$ taking 3, 66, 11 states respectively. These variables describe the connection to the server, e.g., we have that $\Omega_{V1} = \{tcp, udp, icmp\}$. Fig. 7 shows the evolution of the distributions of the discrete variables. For simplicity of the display, improbable states in variables with large domains are not shown. In addition, the variables with temporal information in the data stream have been omitted and we consider that each time step is made of 1000 consecutive instances.

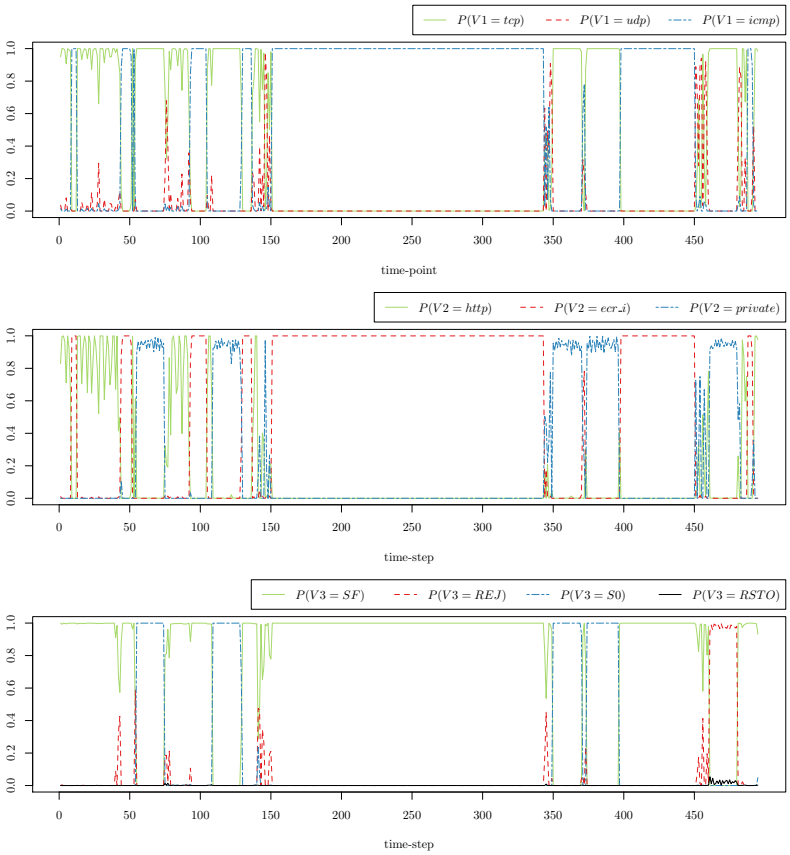

Fig. 7. Evolution of the probability values for the variables $V1$, $V2$ and $V3$. For simplicity, 63 improbable states of $V2$ and 7 of $V3$ are not shown.

The model used for this data stream is shown in Fig. 8. The first group of probabilities (or states) that we consider are $P(V1 = tcp)$, $P(V1 = icmp)$ and

[5] `http://www.liaad.up.pt/kdus/downloads/kdd-cup-10-percent-dataset`

$P(V2 = http)$. The usual traffic in the server are HTTP packages which are sent using TCP protocol. On the other hand, during a denegation of service attack, the number of ICMP packages increases. For that reason, it is interesting to concept drift in these 3 states. This is done with the hidden variable $H1^t$. Then, the hidden variable $H2^t$ detects the concept in $P(V3 = REJ)$, $P(V3 = RSTO)$. For most of the time steps, these two states are not probable.

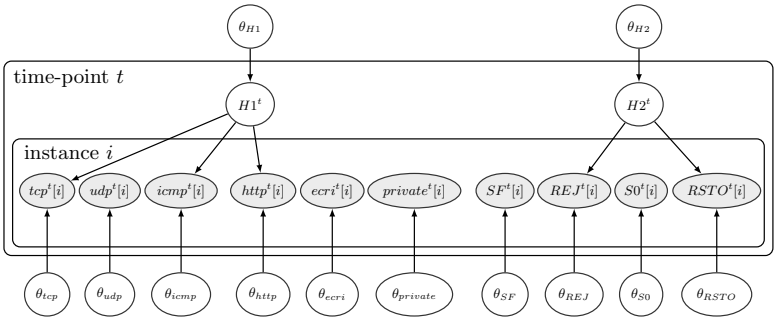

Fig. 8. Proposed model for concept drift detection in the intrusion data stream.

Fig. 9 shows the output of the previous model. We can observe that the changes in the distributions of the states $tcp, icmp$ and $http$ imply a change in $H1^t$. This is also a robust method which has a smoothing effect: short changes in the distributions are ignored. For example, in $P(V3 = REJ)$ many probability peaks appear in a few time points which are not reflected in $H2^t$.

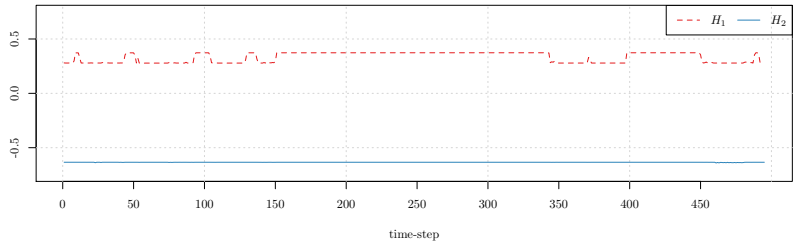

Fig. 9. Results for the intrusion data stream.

5 Conclusions

In this paper we have presented a method for detecting changes in the underlying distributions (i.e., concept drift) of discrete streaming data. Our approach is based on the use of conditional linear Gaussian Bayesian networks. With this approach, we can detect changes in the probabilities associated to subsets of the variable domains. In the experimental work, we have seen that our method can be applied to large and high dimensional data streams.

Acknowledgements

Authors have been jointly supported by the Spanish Ministry of Economy and Competitiveness and by the European Regional Development Fund (FEDER) under the projects TIN2013-46638-C3-2-P, TIN2015-74368-JIN and TIN2016-77902-C3- 2-P.

References

1. H. Borchani, A. M. Martínez, A. R. Masegosa, H. Langseth, T. D. Nielsen, A. Salmerón, A. Fernández, Anders L. Madsen, and R. Sáez. Modeling concept drift: A probabilistic graphical model based approach. In *International Symposium on Intelligent Data Analysis*, pages 72–83. Springer, 2015.
2. T. Broderick, N. Boyd, A. Wibisono, A. C. Wilson, and M. I. Jordan. Streaming variational Bayes. In *Advances in Neural Information Processing Systems*, pages 1727–1735, 2013.
3. W. L. Buntine. Operations for learning with graphical models. *JAIR*, 2:159–225, 1994.
4. R. Cabañas, A. M. Martínez, A. R. Masegosa, D. Ramos-López, A. Samerón, T. D. Nielsen, H. Langseth, and A. L. Madsen. Financial data analysis with PGMs using AMIDST. In *Data Mining Workshops (ICDMW), 2016 IEEE 16th International Conference on*, pages 1284–1287. IEEE, 2016.
5. J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, 46(4):44, 2014.
6. F. V. Jensen and T. D. Nielsen. *Bayesian Networks and Decision Graphs*. Springer-Verlag, Berlin, Germany, 2007.
7. S. L. Lauritzen. Propagation of probabilities, means, and variances in mixed graphical association models. *Journal of the American Statistical Association*, 87(420):1098–1108, 1992.
8. S. L. Lauritzen. *Graphical Models*. Oxford University Press, 1996.
9. A. R. Masegosa, A. M. Martinez, and H. Borchani. Probabilistic graphical models on multi-core cpus using java 8. *IEEE Computational Intelligence Magazine*, 11(2):41–54, 2016.
10. A. R. Masegosa, A. M. Martínez, H. Langseth, T. D. Nielsen, A. Salmerón, D. Ramos-López, and A. L. Madsen. d-VMP: Distributed variational message passing. In *JMLR: Proceedings of the 8th International Conference on Probabilistic Graphical Models*, pages 321–332, 2016.
11. K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.
12. J. Pearl. *Bayesian networks: A model of self-activated memory for evidential reasoning*. University of California. Computer Science Department, 1985.
13. J. C. Schlimmer and R. H. Granger. Incremental learning from noisy data. *Machine Learning*, 1(3):317–354, 1986.
14. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani. A detailed analysis of the KDD CUP 99 data set. In *Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on*, pages 1–6. IEEE, 2009.
15. G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23(1):69–101, 1996.
16. J. M. Winn and C. M. Bishop. Variational message passing. *Journal of Machine Learning Research*, 6:661–694, 2005.