Conditional Log-Likelihood for Continuous Time Bayesian Network Classifiers

Daniele Codecasa and Fabio Stella

DISCo, Università degli Studi di Milano-Bicocca,
Viale Sarca 336, 20126 Milano, Italy
{codecasa,stella}@disco.unimib.it

Abstract. Continuous time Bayesian network classifiers are designed for analyzing multivariate streaming data when time duration of events matters. New continuous time Bayesian network classifiers are introduced while their conditional log-likelihood scoring function is developed. A learning algorithm, combining conditional log-likelihood with Bayesian parameter estimation is developed. Classification accuracy values achieved on synthetic and real data by continuous time and dynamic Bayesian network classifiers are compared. Numerical experiments show that the proposed approach outperforms dynamic Bayesian network classifiers and continuous time Bayesian network classifiers learned with log-likelihood.

Keywords: Multivariate streaming data, conditional log-likelihood.

1 Introduction

Streaming data are relevant to *finance*, with reference to high frequency trading [4], *computer science*, with reference to system error logs, web search query logs, network intrusion detection, social networks [23] and temporal semantic [15], and *engineering*, with reference to image, audio and video processing [30]. They are also important for analyzing GPS data as shown in [14] and [3] where buses and animals paths are analyzed. Streaming data are becoming increasingly important in medicine for patient monitoring and continuous time diagnosis including the study of firing pattern of neurons [27]. Finally, they are becoming relevant in biology where time course data [1] allow the reconstruction of gene regulatory networks, to model the evolution of infections, and to learn and analyze metabolic networks [28].

Dynamic Bayesian networks (DBNs) [5] and hidden Markov models (HMMs) [19] offer a natural way to represent and analyze streaming data. However, DBNs are concerned with discrete time and thus suffer from several limitations due to the fact that it is not clear how timestamps should be discretized. In the case where a too slow sampling rate is used the data will be poorly represented while a too fast sampling rate rapidly makes learning and inference prohibitive. Furthermore, it has been pointed out [12] that when allowing long term dependencies it is required to condition on multiple steps into the past, and thus choosing a too fast sampling rate increases the number of such steps that need to be conditioned on.

Continuous time Bayesian networks (CTBNs) [16], continuous time noisy-or (CT-NOR) [22], Poisson cascades [23] and Poisson networks [20] together with the piecewise-constant conditional intensity model (PCIM) [12] are interesting models to represent and analyze continuous time processes. CT-NOR and Poisson cascades are devoted to model event streams while they require the modeler to specify a parametric form for temporal dependencies. This aspect significantly impacts performance and the problem of model selection in CT-NOR and Poisson cascades has not been addressed yet. This limitation is overcame by PCIMs which perform structure learning to model how events in the past affect future events of interest. CTBNs are homogeneous Markov models which allow to represent joint trajectories of discrete finite variables.

In this paper we consider the problem of *temporal classification*, where data stream measurements are available over a period of time in history while the class is expected to occur in the future. This kind of problem can be addressed by discrete and continuous time models. Discrete time models include dynamic latent classification models [31], a specialization of the latent classification model (LCM) [13], and DBNs [5]. Continuous time models, as continuous time Bayesian network classifiers (CTBNCs) [24], overcame the problem of timestamps discretization. The main contributions of the paper are:

- definition of new classifiers from the class of CTBNCs,
- development of the conditional log-likelihood scoring function for CTBNCs,
- performance comparison of CTBNCs learned with the conditional log-likelihood score to CTBNCs learned with log-likelihood score and to DBN classifiers.

The paper is organized as follows; Section 2 is devoted to notations and definitions. New classifiers are introduced and analyzed in Section 3. Section 4 concerns numerical experiments where synthetic datasets generated from models of increasing complexity are used. In this section a real dataset on post-stroke rehabilitation is analyzed. Finally, Section 5 is devoted to comments.

2 Continuous time classification

2.1 Continuous Time Bayesian Networks

Dynamic Bayesian networks (DBNs) model dynamic systems without representing time explicitly. They discretize time to represent a dynamic system through several time slices. In [17] the authors pointed out that "*since DBNs slice time into fixed increments, one must always propagate the joint distribution over the variables at the same rate*" . Therefore, if the system consists of processes which evolve at different time granularities and/or the obtained observations are irregularly spaced in time, the inference process may become computationally intractable.

Continuous time Bayesian networks (CTBNs) overcome the limitations of DBNs by explicitly representing temporal dynamics and thus allow us to recover the probability distribution over time when specific events occur. CTBNs have been used to discover intrusion in computers [29], to analyze the reliability of dynamic systems [2], for learning social networks dynamics [8] and to model cardiogenic heart failure [10]. A continuous time Bayesian network (CTBN) is a probabilistic graphical model whose nodes are associated with random variables and whose state evolves continuously over time.

Definition 1. *(Continuous time Bayesian network). [17]. Let* $\mathbf{X}$ *be a set of random variables* $X_1, X_2, ..., X_N$*. Each* X_n *has a finite domain of values* $Val(X_n) = \{x_1, x_2, ..., x_I\}$*. A continuous time Bayesian network* $\aleph$ *over* $\mathbf{X}$ *consists of two components: the first is an initial distribution* $P^0_{\mathbf{X}}$*, specified as a Bayesian network* $\mathcal{B}$ *over* $\mathbf{X}$*. The second is a continuous transition model, specified as:*

- *a directed (possibly cyclic) graph* $\mathcal{G}$ *whose nodes are* $X_1, X_2, ..., X_N$*;* $Pa(X_n)$ *denotes the parents of* X_n *in* $\mathcal{G}$*.*
- *a conditional intensity matrix,* $\mathbf{Q}^{Pa(X_n)}_{X_n}$*, for each variable* $X_n \in \mathbf{X}$*.*

Given the random variable X_n, the *conditional intensity matrix* (CIM) $\mathbf{Q}^{Pa(X_n)}_{X_n}$ consists of a set of intensity matrices, one intensity matrix

$$\mathbf{Q}^{pa(X_n)}_{X_n} = \begin{bmatrix} -q^{pa(X_n)}_{x_1} & q^{pa(X_n)}_{x_1x_2} & . & q^{pa(X_n)}_{x_1x_I} \\ q^{pa(X_n)}_{x_2x_1} & -q^{pa(X_n)}_{x_2} & . & q^{pa(X_n)}_{x_2x_I} \\ . & . & . & . \\ q^{pa(X_n)}_{x_Ix_1} & q^{pa(X_n)}_{x_Ix_2} & . & -q^{pa(X_n)}_{x_I} \end{bmatrix},$$

for each instantiation $pa(X_n)$ of the parents $Pa(X_n)$ of node X_n, where $q^{pa(X_n)}_{x_i} = \sum\limits_{x_j \neq x_i} q^{pa(X_n)}_{x_ix_j}$ is the rate of leaving state x_i for a specific instantiation $pa(X_n)$ of $Pa(X_n)$, while $q^{pa(X_n)}_{x_ix_j}$ is the rate of arriving to state x_j from state x_i for a specific instantiation $pa(X_n)$ of $Pa(X_n)$. Matrix $\mathbf{Q}^{pa(X_n)}_{X_n}$ can equivalently be summarized by using two types of parameters, $q^{pa(X_n)}_{x_i}$ which is associated with each state x_i of the variable X_n when its parents are set to $pa(X_n)$, and $\theta^{pa(X_n)}_{x_ix_j} = \frac{q^{pa(X_n)}_{x_ix_j}}{q^{pa(X_n)}_{x_i}}$ which represents the probability of transitioning from state x_i to state x_j, when it is known that the transition occurs at a given instant in time.

Example 1. Figure 1 shows a part of the drug network introduced in [17]. It contains a cycle, indicating that whether a person is hungry (H) depends on how full his/her stomach (S) is, which depends on whether or not he/she is eating (E), which in turn depends on whether he/she is hungry. We assume that E and H are binary variables (i.e. no (n)/yes (y)) while S is ternary (i.e. full (f)/average (a)/empty (e)). Then, the CIMs for E are the [2x2] matrices $\mathbf{Q}^n_E$, and $\mathbf{Q}^y_E$, the CIMs for S are the [3x3] matrices $\mathbf{Q}^n_S$ and $\mathbf{Q}^y_S$, while the CIMs for H are the [2x2] matrices $\mathbf{Q}^f_H$, $\mathbf{Q}^a_H$ and, $\mathbf{Q}^e_H$. For matters of brevity we only show $\mathbf{Q}^y_S$ with two equivalent parametric representations:

$$\mathbf{Q}^y_S = \begin{bmatrix} -q^y_f & q^y_{fa} & q^y_{fe} \\ q^y_{af} & -q^y_a & q^y_{ae} \\ q^y_{ef} & q^y_{ea} & -q^y_e \end{bmatrix} = \begin{bmatrix} -.03 & .02 & .01 \\ 5.99 & -6 & .01 \\ 1 & 5 & -6 \end{bmatrix} \tag{1}$$

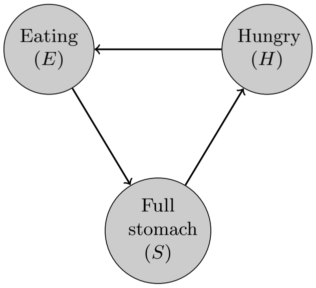

Fig. 1. A part of the drug network.

$$\mathbf{Q}_I^y = \begin{bmatrix} q_f^y & 0 & 0 \\ 0 & q_a^y & 0 \\ 0 & 0 & q_e^y \end{bmatrix} \left(\begin{bmatrix} 0 & \theta_{fa}^y & \theta_{fe}^y \\ \theta_{af}^y & 0 & \theta_{ae}^y \\ \theta_{ef}^y & \theta_{ea}^y & 0 \end{bmatrix} - \mathbf{I} \right) = \begin{bmatrix} .03 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 6 \end{bmatrix} \left(\begin{bmatrix} 0 & \frac{.02}{.03} & \frac{.01}{.03} \\ \frac{5.99}{6} & 0 & \frac{.01}{6} \\ \frac{1}{6} & \frac{5}{6} & 0 \end{bmatrix} - \mathbf{I} \right) \quad (2)$$

where $\mathbf{I}$ is the identity matrix. If we view units of time as hours, then we expect a person who has an empty stomach ($S{=}e$) and is eating ($E{=}y$) to stop having an empty stomach in 10 minutes ($\frac{1}{6}$ hour). The stomach will then transition from state e ($S{=}e$) to state a ($S{=}a$) with probability $\frac{5}{6}$ and to state f ($S{=}f$) with probability $\frac{1}{6}$. Equation 1 is a compact representation of the CIM while Equation 2 is useful because it explicitly represents the transition probability value from state x to state x', i.e. $\theta_{xx'}^{pa(X)}$.

CTBNs allow two types of evidence, namely *point evidence* and *continuous evidence*, while HMMs and DBNs allow only point evidence. *Continuous evidence* is the knowledge of the states of a set of variables $\mathbf{X}$ throughout an entire half-closed interval of time $[t_1, t_2)$: $\mathbf{Z}^{[t_1,t_2)} = \mathbf{z}^{[t_1,t_2)}$, where $\mathbf{Z}^{[t_1,t_2)} = (X_1^{[t_1,t_2)}, X_2^{[t_1,t_2)}, ..., X_k^{[t_1,t_2)})$ while $\mathbf{z}^{[t_1,t_2)} = (x_1^{[t_1,t_2)}, x_2^{[t_1,t_2)}, ..., x_k^{[t_1,t_2)})$.

Inference in CTBNs can be performed by exact and approximate algorithms. *Full amalgamation* [17] allows exact inference by generating an exponentially-large matrix representing the transition model over the entire state space. Exact inference in CTBNs is known to be NP-hard, and thus different approximate algorithms have been proposed. In [16] the authors introduced the *Expectation Propagation* algorithm (EP), while in [21] an optimized variant of EP is presented. Alternatives are offered by sampling based inference algorithms such as *importance sampling* algorithm [7] and *Gibbs sampling* algorithm [6].

Given the dataset $\mathcal{D}$, parameter learning is similar to *maximum likelihood estimation*, but accounts for the *imaginary counts* $\alpha_x^{pa(X)}$, $\alpha_{xx'}^{pa(X)}$ and, $\tau_x^{pa(X)}$ of the hyperparameters:

$$q_x^{pa(X)} = \frac{\alpha_x^{pa(X)} + M[x \mid pa(X)]}{\tau_x^{pa(X)} + T[x \mid pa(X)]}; \; \theta_{xx'}^{pa(X)} = \frac{\alpha_{xx'}^{pa(X)} + M[x, x' \mid pa(X)]}{\alpha_x^{pa(X)} + M[x \mid pa(X)]} \quad (3)$$

where $M[x, x' \mid pa(X)]$, $M[x \mid pa(X)]$ and $T[x \mid pa(X)]$ are the *sufficient statistics*. $M[x, x' \mid pa(X)]$ is the count of transitions from state x to state x' for node X when the state of its parents $Pa(X)$ is set to $pa(X)$. $M[x \mid pa(X)] = \sum_{x' \neq x} M[x, x' \mid pa(X)]$ is the count of transitions leaving state x of node X when the state of its parents $Pa(X)$ is set to $pa(X)$. Finally, $T[x \mid pa(X)]$ represents the time spent in state x by the variable X when the state of its parents $Pa(X)$ is set to $pa(X)$.

Learning the structure of a CTBN from a given dataset $\mathcal{D}$ has been addressed as an optimization problem over possible CTBN structures [18]. It consists of finding the structure $\mathcal{G}$ which maximizes the following score:

$$\mathbf{score}_{\aleph}\left(\mathcal{G} : \mathcal{D}\right) = \ln P(\mathcal{D}|\mathcal{G}) + \ln P(\mathcal{G}). \tag{4}$$

However, the search space of this optimization problem is significantly simpler than that of BNs or DBNs. Indeed, it is known that learning the optimal structure of a BN is NP-hard, while the same does not hold true in the context of CTBNs where all edges are across time and thus represent the effect of the current value of one variable on the next value of the other variables. Therefore, no acyclicity constraints arise, and it is possible to optimize the parent set for each variable of the CTBN independently.

2.2 Continuous Time Bayesian Network Classifiers

Continuous time Bayesian network classifiers (CTBNCs) [24] are a specialization of CTBNs. They allow polynomial time classification inference which is NP-hard for general CTBNs. Classifiers from this class explicitly represent the evolution in continuous time of the set of random variables X_n, $n = 1, 2, ..., N$ which are assumed to depend on the class node Y.

Definition 2. *(Continuous time Bayesian network classifier)*[1]. *A continuous time Bayesian network classifier is a pair* $\mathcal{C} = \{\aleph, P(Y)\}$ *where* $\aleph$ *is a CTBN model with attribute nodes* $X_1, X_2, ..., X_N$, Y *is the class node with marginal probability* $P(Y)$ *on states* $Val(Y) = \{y_1, y_2, ..., y_K\}$, $\mathcal{G}$ *is the graph of the CTBNC, such that the following conditions hold:*

- $Pa(Y) = \emptyset$, *the class variable* Y *is associated with a root node;*
- Y *is fully specified by* $P(Y)$ *and does not depend on time.*

Given a dataset $\mathcal{D}$ with no missing data, a CTBNC is learned by maximizing the score (4) subjected to the constraints listed in Definition 2. However, exact learning requires to set in advance the maximum number of parents k for the nodes $X_1, X_2, ..., X_N$ [16]. Therefore, in the case where k is not small a considerable computational effort is required to find the graph structure $\mathcal{G}^*$ maximizing the score (4). In such a case we resort to hill-climbing or to the continuous time naive Bayes classifier.

[1] This definition differs from the one proposed in [24]. In fact, we do not require the CTBNC graph to be connected. Thus, features selection is obtained as by product of CTBNC structural learning.

Definition 3. *(Continuous time naive Bayes classifier). [24] A continuous time naive Bayes classifier is a continuous time Bayesian network classifier $\mathcal{C} = \{\aleph, P(Y)\}$ such that $Pa(X_n) = \{Y\}$, $n = 1, 2, ..., N$.*

According to [24] a CTBNC $\mathcal{C} = \{\aleph, P(Y)\}$ classifies a stream of continuous time evidence $\mathbf{z} = (x_1, x_2, ..., x_N)$ for the attributes $\mathbf{Z} = (X_1, X_2, ..., X_N)$ over J contiguous time intervals, i.e. a stream of continuous time evidence $\mathbf{Z}^{[t_1,t_2)} = \mathbf{z}^{[t_1,t_2)}$, $\mathbf{Z}^{[t_2,t_3)} = \mathbf{z}^{[t_2,t_3)}$, $\ldots$, $\mathbf{Z}^{[t_{J-1},t_J)} = \mathbf{z}^{[t_{J-1},t_J)}$, by selecting the value y^* for the class Y which maximizes the posterior probability $P(Y|\mathbf{z}^{[t_1,t_2)}, \mathbf{z}^{[t_2,t_3)}, ..., \mathbf{z}^{[t_{J-1},t_J)})$, which is proportional to

$$P(Y) \prod_{j=1}^{J} q_{x_{m_j}^j x_{m_j}^{j+1}}^{pa(X_{m_j})} \prod_{n=1}^{N} exp\left(-q_{x_n^j}^{pa(X_n)} \delta_j\right), \tag{5}$$

where:

- $\delta_j = t_j - t_{j-1}$ is the length of the j^{th} time interval of the stream $\mathbf{z}^{[t_1,t_2)}$, $\mathbf{z}^{[t_2,t_3)}$, ..., $\mathbf{z}^{[t_{J-1},t_J)}$ of continuous time evidence;
- $q_{x_n^j}^{pa(X_n)}$ is the parameter associated with state x_n^j, in which the variable X_n was during the j^{th} time interval, given the state of its parents $pa(X_n)$ during the j^{th} time intervals;
- $q_{x_m^j x_m^{j+1}}^{pa(X_m)}$ is the parameter associated with the transition from state x_m^j, in which the variable X_m was during the j^{th} time interval, to state x_m^{j+1}, in which the variable X_m will be during the $(j+1)^{th}$ time interval, given the state of its parents $pa(X_m)$ during the j^{th} and the $(j+1)^{th}$ time intervals.

Learning algorithm based on log-likelihood score for the CTNB and inference algorithm for the class of CTBNCs are described in [24].

3 Max-*k* Classifiers

3.1 Definitions

Structural learning for CTBNs is a polynomial time problem with respect to the maximum number of parents k. Nevertheless, increasing k, rapidly brings to considerable computational efforts while implies more data is necessary to learn the node's parameters values conditioned on possible parents' instantiations. To overcome this limitations we propose the following instances from the class of CTBNCs; the Max-k Augmented CTNB (Max-k ACTNB) and the Max-k CTBNC (Max-k CTBNC).

Definition 4. *(Max-k Continuous Time Bayesian Network Classifier). A max-k continuous time Bayesian network classifier is a couple $\mathcal{M} = \{\mathcal{C}, k\}$, where $\mathcal{C}$ is a continuous time Bayesian network classifier $\mathcal{C} = \{\aleph, P(Y)\}$ such that the number of parents $|Pa(X_n)|$ for each attribute node X_n is bounded by a positive integer k. Formally, the following condition holds; $|Pa(X_n)| \leq k$, $n = 1, 2, ..., N$, $k \geq 0$.*

Definition 5. *(Max-k Augmented Continuous Time Naive Bayes). A max-k augmented continuous time naive Bayes classifier is a max-k continuous time Bayesian network classifier such that the class node Y belongs to the parents set of each attribute node X_n, $n = 1, 2, ..., N$. Formally, the following condition holds; $Y \in Pa(X_n)$, $n = 1, 2, ..., N$.*

ACTNB constraints the class variable Y to be a parent of each node X_n, $n = 1, 2, ..., N$. In this way it tries to compensate for relevant dependencies between nodes which could be excluded to satisfy the constraint on the maximum number of parents k.

3.2 Learning

Learning a CTBNC from data consists of learning a CTBN where a specific node, i.e. the class node Y, does not depend on time. In such a case, the learning algorithm runs, for each attribute node X_n, $n = 1, 2, ..., N$, a local search procedure to find its optimal set of parents, i.e. the set of parents which maximizes a given score function. Furthermore, for each attribute node X_n, $n = 1, 2, ..., N$, no more than k parents are selected. The structural learning algorithm proposed in [18] uses a score function (4) based on log-likelihood. This algorithm can be easily adapted to learn a CTBNC by introducing the constraint that the class node Y must not depend on time.

3.3 Log-likelihood and Conditional Log-likelihood

Log-likelihood is not the only scoring function which can be used to learn the structure of a CTBN classifier. Following what presented and discussed in [9], the log-likelihood function:

$$LL(\mathcal{M} \mid \mathcal{D}) = \sum_{i=1}^{|\mathcal{D}|} \log P_{\aleph}(y_i \mid \mathbf{x}_i^1, ..., \mathbf{x}_i^{J_i}) + \log P_{\aleph}(\mathbf{x}_i^1, ..., \mathbf{x}_i^{J_i}). \qquad (6)$$

consists of two components; $\log P_{\aleph}(y_i \mid \mathbf{x}_i^1, ..., \mathbf{x}_i^{J_i})$, which measures the classification *capability* of the model, and $\log P_{\aleph}(\mathbf{x}_i^1, ..., \mathbf{x}_i^{J_i})$, which models the dependencies between the nodes.

In [9] the authors remarked that in the case where the number of the attribute nodes X_n, $n = 1, 2, ..., N$ is large, the contribution, to the scoring function value (6), of $\log P_{\aleph}(\mathbf{x}_i^1, ..., \mathbf{x}_i^{J_i})$ overwhelms the contribution of $\log P_{\aleph}(y_i \mid \mathbf{x}_i^1, ..., \mathbf{x}_i^{J_i})$. However, the contribution of $\log P_{\aleph}(\mathbf{x}_i^1, ..., \mathbf{x}_i^{J_i})$ is not directly related to the classification accuracy achieved by the classifier. Therefore, to improve the classification performance, in [9] it has been suggested to use the *conditional log-likelihood* as scoring function. In such a case the maximization of the conditional log-likelihood results in maximizing the classification performance of the model without paying specific attention to the discovery of the existing dependencies between the attribute nodes X_n, $n = 1, 2, ..., N$.

The conditional log-likelihood of the CTBNC, which can be written as follows:

$$CLL(\mathcal{M} \mid \mathcal{D}) = \sum_{i=1}^{|\mathcal{D}|} \log P_{\aleph}(y_i \mid \mathbf{x}_i^1, ..., \mathbf{x}_i^{J_i}) = \sum_{i=1}^{|\mathcal{D}|} \log \left(\frac{P_{\aleph}(\mathbf{x}_i^1, ..., \mathbf{x}_i^{J_i} \mid y_i) P_{\aleph}(y_i)}{P_{\aleph}(\mathbf{x}_i^1, ..., \mathbf{x}_i^{J_i})} \right)$$
$$= \sum_{i=1}^{|\mathcal{D}|} \log \left(P_{\aleph}(y_i)\right) + \log \left(P_{\aleph}(\mathbf{x}_i^1, ..., \mathbf{x}_i^{J_i} \mid y_i) \right) - \log \left(\sum_{y'} P_{\aleph}(y') P_{\aleph}(\mathbf{x}_i^1, ..., \mathbf{x}_i^{J_i} \mid y') \right). \tag{7}$$

consists of *class probability* (8), *posterior probability* (9), and *denominator* (10) terms. The *class probability* term is estimated from the dataset $\mathcal{D}$ as follows:

$$\sum_{i=1}^{|\mathcal{D}|} \log \left(P_{\aleph}(y_i)\right) = \sum_{y} M[y] \log(\theta_y) \tag{8}$$

where θ_y represents the parameter associated with the probability of class y.

From (5) it is possible to write the following:

$$P_{\aleph}(\mathbf{x}^1, ..., \mathbf{x}^J \mid y) = \prod_{j=1}^{J} q_{x_{m_j}^j x_{m_j}^{j+1}}^{pa(X_{m_j})} \prod_{n=1}^{N} exp \left(-q_{x_n^j}^{pa(X_n)} \delta_j \right)$$
$$= \prod_{j=1}^{J} q_{x_{m_j}^j}^{pa(X_{m_j})} \theta_{x_{m_j}^j x_{m_j}^{j+1}}^{pa(X_{m_j})} \prod_{n=1}^{N} exp \left(-q_{x_n^j}^{pa(X_n)} \delta_j \right)$$

Therefore, the *posterior probability* term is estimated as follows:

$$\textstyle\sum_{i=1}^{|\mathcal{D}|} \log \left(P_{\aleph}(\mathbf{x}_i^1, ..., \mathbf{x}_i^{J_i} \mid y_i) \right) = \displaystyle\sum_{n=1}^{N} \sum_{x_n, pa(X_n)} M[x_n \mid pa(X_n)] \log \left(q_{x_n}^{pa(X_n)} \right)$$
$$- q_{x_n}^{pa(X_n)} T[x_n \mid pa(X_n)] + \sum_{x'_n \neq x_n} M[x_n x'_n \mid pa(X_n)] \log(\theta_{x_n x'_n}^{pa(X_n)}). \tag{9}$$

The *denominator* term, because of the sum, can not be decomposed further. The sufficient statistics allow us to write the following:

$$\textstyle\sum_{i=1}^{|\mathcal{D}|} \log \left(\displaystyle\sum_{y'} P_{\aleph}(y') P_{\aleph}(\mathbf{x}_i^1, ..., \mathbf{x}_i^{J_i} \mid y') \right) = \tag{10}$$

$$= \log \left(\sum_{y'} \theta_{y'} \prod_{n=1}^{N} \prod_{x_n, pa'(X_n)} (q_{x_n}^{pa'(X_n)})^{M[x_n|pa'(X_n)]} \exp(-q_{x_n}^{pa'(X_n)} T[x_n \mid pa'(X_n)]) \right.$$
$$\left. \prod_{x'_n \neq x_n} (\theta_{x_n x'_n}^{pa'(X_n)})^{M[x_n x'_n|pa'(X_n)]} \right)$$

where $pa(X_n) = \{\pi_n \cup y\}$, $pa'(X_n) = \{\pi_n \cup y'\}$, while π_n is the instantiation of the non-class parents of the attribute node X_n.

Unfortunately, no closed form solution exists to compute the optimal value of the model's parameters, i.e. those parameters values which maximize the conditional log-likelihood (7). Therefore, the approach introduced and discussed in [11] is followed. The scoring function is computed by using the conditional log-likelihood, while parameters values are obtained by using the Bayesian approach (3).

4 Numerical experiments

The performance of CTBNCs, namely CTNB, K=2 ACTNB, K=2 CTBNC, K=3 CTBNC, and K=4 CTBNC, is compared to that of DBNs by exploiting synthetic datasets. Classifiers are associated with a suffix related to the scoring function which has been used for learning. Suffix LL is associated with log-likelihood scoring while suffix CLL is associated with conditional log-likelihood scoring. To fairly compare conditional log-likelihood score to log-likelihood score, no graph's structure penalization terms have been added to the two score functions. Numerical experiments for performance estimation and comparison of classifiers are implemented with 10 folds cross validation.

4.1 Synthetic datasets

Accuracy, learning and inference time of different CTBNCs are compared on synthetic datasets generated by sampling from models of increasing complexity. Datasets consist of $1,000$ trajectories with average length ranging from 300 (CTNBs) to 1,400 (K=4 CTBNCs). Analyzed model structures are CTNB, K=2 ACTNB, K=2 CTBNC, K=3 CTBNC, and K=4 CTBNC. For each structure, different assignments of parameters values (q parameters) are sampled in a given interval. Each pair, (*structure*, *parameters assignment*), is used to generate a learning dataset. Performance is analyzed on *full datasets* (100%) and on *reduced datasets*, i.e. when the number and the length of trajectories are reduced to: 80%, 60%, 40%, and 20%. Accuracy values on *full datasets* (100% datasets) are summarized in Table 1 while Figure 2 depicts how the tested models behave when reduced datasets (80%, 60%, 40%, 20%) are used for learning.

DBNs are outperformed by all continuous time models while CLL scoring seems to perform better or at least to be never inferior than LL scoring. Figure 2 shows that CLL scoring strongly outperforms LL scoring when the amount of data is limited. This is probably due to the effectiveness of CLL scoring to discover weak dependencies between variables and thus to its tendency to add the class variable as a parent of all nodes useful for the classification task. On the contrary, when the amount of data is too low, CLL scoring tends to overfit by learning classifiers which are too complex. In these cases, LL scoring achieves poor accuracy too, while the CTNB is the best option (see Fig. 2).

Test	CTNB	$k = 2$ ACTNB (LL)	$k = 2$ ACTNB (CLL)	$k = 2$ CTBNC (LL)	$k = 2$ CTBNC (CLL)	$k = 3$ CTBNC (LL)	$k = 3$ CTBNC (CLL)	$k = 4$ CTBNC (LL)	$k = 4$ CTBNC (CLL)	DBN- NB1	DBN- NB2
CTNB	**0.95**	**0.95**	0.93	0.93	0.92	0.93	0.82	0.93	0.64	0.80	0.81
K2ACTNB	0.78	0.89	**0.92**	0.76	**0.92**	0.76	0.85	0.76	0.72	0.62	0.63
K2CTBNC	0.68	0.84	**0.86**	**0.85**	**0.86**	**0.85**	0.76	**0.85**	0.60	0.48	0.50
K3CTBNC	0.49	0.65	0.63	0.66	0.63	**0.79**	0.75	**0.79**	0.64	0.32	0.33
K4CTBNC	0.64	0.74	0.79	0.69	0.79	0.76	**0.94**	0.79	0.90	0.40	0.40

Table 1. Classifier's average accuracy value with respect to different categories of the dataset generating model, 10 folds cross validation over *full datasets* (100%). Bolded characters are associated with the best model with 90% of confidence.

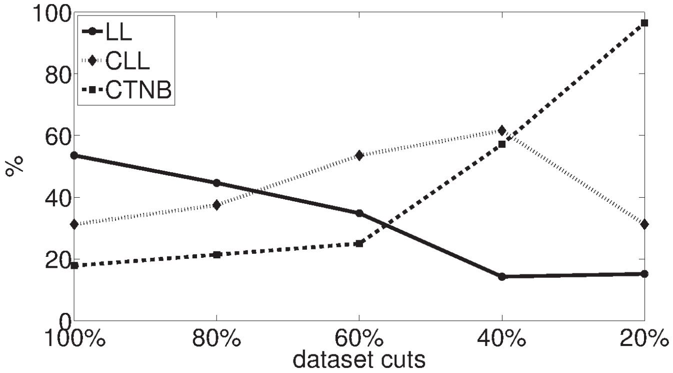

Fig. 2. Percentage of numerical experiments where LL (CLL) achieved a better accuracy value than the one achieved by CLL (LL) with 90% of confidence. Percentage of numerical experiments where CTNB achieved a better/comparable accuracy value than the best accuracy achieved by any continuous time model (bold dotted line) .

Inference times on continuous time models are comparable, while inference time required by DBNs make them impractical. Structural learning of CTBNCs with log-likelihood scoring is slightly faster than with conditional log-likelihood scoring (tables and figures not shown for the sake of brevity).

4.2 Post-stroke rehabilitation dataset

In [26] the authors proposed a movement recognition system to face automatic post-stroke rehabilitation problem. The idea is to provide the patient with a system capable to recognize the movements and to inform him/her about the correctness of the performed rehabilitation exercise. The authors focused on upper limb post-stroke rehabilitation and provided a dataset of 7 rehabilitation exercises. For each exercise 120 multivariate trajectories are recorded by using 29 sensors working with a frequency of 30 Hz [25]. Each movement is addressed separately as classification problem. We focus the attention on 2, and 6 classes problems where classes are associated with the same number of trajectories.

			K=2	K=2	K=2	K=2	K=3	K=3	K=4	K=4
# classes	Measure	CTNB	ACTNB	ACTNB	CTBNC	CTBNC	CTBNC	CTBNC	CTBNC	CTBNC
			(LL)	(CLL)	(LL)	(CLL)	(LL)	(CLL)	(LL)	(CLL)
2 classes	Accuracy	0.98	0.97	**0.99**	0.87	0.85	0.87	0.92	0.87	0.95
	Precision	0.97	0.97	**0.99**	0.86	0.85	0.86	0.93	0.86	0.95
	Recall	0.98	0.98	**0.99**	0.88	0.84	0.88	0.92	0.88	0.96
6 classes	Accuracy	**0.91**	**0.91**	**0.89**	0.81	**0.88**	0.81	**0.88**	0.81	**0.88**
	Precision	**0.92**	**0.91**	**0.89**	0.84	**0.89**	0.84	**0.89**	0.84	**0.90**
	Recall	**0.90**	**0.90**	**0.88**	0.81	**0.88**	0.82	**0.88**	0.82	**0.89**

Table 2. Average accuracy, precision and, recall for the post-stroke rehabilitation dataset (10 folds CV). Bolded characters indicate the best models with 90% of confidence.

Accuracy, precision and recall values achieved by almost all CLL classifiers are better than the values achieved by their LL counterparts. For the 6 classes classification problem, in the case where no information about variables' dependency is available, CLL outperforms LL (Table 2). K=2 ACTNB, learned with CLL scoring, implements the optimal trade-off between time and accuracy. Indeed, for both 2 and 6 classes classification problems, the K=2 ACTNB model when learned with CLL, achieves the highest accuracy value and is the fastest to learn, because of the small value of the bound on the number of parents. It is worthwhile to mention that CTBNCs when learned with LL scoring are not capable to solve the 6 classes classification problem for many assignments of the priors values. Indeed, it was necessary to evaluate many priors assignments to achieve an acceptable performance value with LL scoring. On the contrary, CLL scoring seems to be very robust with respect to priors assignment.

5 Conclusions

A conditional log-likelihood scoring function has been developed to learn continuous time Bayesian network classifiers. A learning algorithm for CTBNCs has been designed by combining conditional log-likelihood scoring with Bayesian parameter learning. New classifiers models from the class of CTBNCs have been introduced. Numerical experiments, on synthetic and real world streaming datasets, confirm the effectiveness of the proposed approach for CTBNCs learning. Conditional log-likelihood scoring outperforms log-likelihood scoring and DBNs in terms of accuracy. This behavior becomes more and more evident as the amount of the available streaming data become scarce.

Acknowledgements

The authors would like to acknowledge the many helpful suggestions of the anonymous reviewers which helped to improve the paper clarity and quality. The authors would like to thank Project Automation S.p.A. for funding the Ph.D. programme of Daniele Codecasa.

References

1. Barber, D., Cemgil, A.: Graphical models for time-series. Signal Processing Magazine, IEEE 27(6), 18–28 (2010)
2. Boudali, H., Dugan, J.: A continuous-time bayesian network reliability modeling, and analysis framework. IEEE Trans. on Reliability 55(1), 86–97 (2006)
3. Costa, G., Manco, G., Masciari, E.: Effectively grouping trajectory streams. In: Proc. of the Workshop on New Frontiers in Mining Complex Patterns. pp. 149–151 (2012)
4. Dacorogna, M.: An introduction to high-frequency finance. AP (2001)
5. Dean, T., Kanazawa, K.: A model for reasoning about persistence and causation. Computational Intelligence 5(2), 142–150 (1989)
6. El-Hay, T., Friedman, N., Kupferman, R.: Gibbs sampling in factorized continuous-time markov processes. In: McAllester, D.A., Myllym, P. (eds.) Proc. of the 24th Conf. on UAI. pp. 169–178. AUAI (2008)
7. Fan, Y., Shelton, C.: Sampling for approximate inference in continuous time bayesian networks. In: 10th Int. Symposium on Artificial Intelligence and Mathematics (2008)
8. Fan, Y., Shelton, C.: Learning continuous-time social network dynamics. In: Proc. of the 25th Conf. on UAI. pp. 161–168. AUAI (2009)
9. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning 29(2), 131–163 (1997)
10. Gatti, E., Luciani, D., Stella, F.: A continuous time bayesian network model for cardiogenic heart failure. Flexible Services and Manufacturing Journal pp. 1–20 (2011)
11. Grossman, D., Domingos, P.: Learning bayesian network classifiers by maximizing conditional likelihood. In: Proc. of the 21st Int. Conf. on Machine Learning. pp. 361–368. ACM (2004)
12. Gunawardana, A., Meek, C., Xu, P.: A model for temporal dependencies in event streams. In: Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K. (eds.) Advances in Neural Information Processing Systems, pp. 1962–1970 (2011)
13. Langseth, H., Nielsen, T.D.: Latent classification models. Machine Learning 59(3), 237–265 (2005)
14. Masciari, E.: Trajectory clustering via effective partitioning. In: Flexible Query Answering Systems, pp. 358–370. Springer (2009)
15. Nanni, M., Pedreschi, D.: Time-focused clustering of trajectories of moving objects. Journal of Intelligent Information Systems 27(3), 267–289 (2006)
16. Nodelman, U., Koller, D., Shelton, C.: Expectation propagation for continuous time bayesian networks. In: Proc. of the 21st Conf. on UAI. pp. 431–440. Edinburgh, Scotland, UK (July 2005)
17. Nodelman, U., Shelton, C., Koller, D.: Continuous time bayesian networks. In: Proc. of the 18th Conf. on UAI. pp. 378–387. Morgan Kaufmann (2002)
18. Nodelman, U., Shelton, C., Koller, D.: Learning continuous time bayesian networks. In: Proc. of the 19th Conf. on UAI. pp. 451–458. Morgan Kaufmann (2002)
19. Rabiner, L.: A tutorial on hidden markov models and selected applications in speech recognition. Proc. of the IEEE 77(2), 257–286 (1989)
20. Rajaram, S., Graepel, T., Herbrich, R.: Poisson-networks: A model for structured point processes. In: Proc. of the 10th Int. Workshop on Artificial Intelligence and Statistics (2005)
21. Saria, S., Nodelman, U., Koller, D.: Reasoning at the right time granularity. In: UAI. pp. 326–334 (2007)

22. Simma, A., Goldszmidt, M., MacCormick, J., Barham, P., Black, R., Isaacs, R., Mortier, R.: Ct-nor: Representing and reasoning about events in continuous time. In: Proc. of the 24th Conf. on UAI. pp. 484–493. AUAI (2008)
23. Simma, A., Jordan, M.: Modeling events with cascades of poisson processes. In: Proc. of the 26th Conf. on UAI. pp. 546–555. AUAI (2010)
24. Stella, F., Amer, Y.: Continuous time bayesian network classifiers. Journal of Biomedical Informatics 45(6), 1108–1119 (2012)
25. Tormene, P., Giorgino, T.: Upper-limb rehabilitation exercises acquired through 29 elastomer strain sensors placed on fabric. release 1.0 (2008)
26. Tormene, P., Giorgino, T., Quaglini, S., Stefanelli, M.: Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation. Artificial Intelligence in Medicine 45(1), 11–34 (2009)
27. Truccolo, W., Eden, U., Fellows, M., Donoghue, J., Brown, E.: A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. Journal of neurophysiology 93(2), 1074–1089 (2005)
28. Voit, E.: A First Course in Systems Biology. Garland Science: NY (2012)
29. Xu, J., Shelton, C.: Continuous time bayesian networks for host level network intrusion detection. Machine Learning and Knowledge Discovery in Databases pp. 613–627 (2008)
30. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Computing Surveys 38(4) (2006)
31. Zhong, S., Langseth, H., Nielsen, T.D.: Bayesian networks for dynamic classification. Tech. rep., http://idi.ntnu.no/˜shket/dLCM.pdf (2012)