Online Updating of Conditional Linear Gaussian Bayesian Networks

Anders L Madsen anders@hugin.com
HUGIN EXPERT A/S and Department of Computer Science, Aalborg University, Denmark

Kristian G Olesen kgo@cs.aau.dk
Department of Computer Science, Aalborg University, Denmark

Frank Jensen fj@hugin.com
HUGIN EXPERT A/S, Aalborg, Denmark

Per Henriksen pah@rn.dk
Aalborg University Hospital, Aalborg, Denmark

Thomas M Larsen tml@rn.dk
Region Nordjylland, Aalborg, Denmark

Jørn M Møller jmm@rn.dk
Aalborg University Hospital, Aalborg, Denmark

Abstract

This paper presents a method for online updating of conditional distributions in Bayesian network models with both discrete and continuous variables. The method extends known procedures for updating discrete conditional probability distributions with techniques to cope with conditional Gaussian density functions. The method has a solid foundation for known cases and may be generalised by a heuristic scheme for fractional updating when discrete parents are not known. A fading mechanism is described to prevent the system being too conservative as cases accumulate over long time periods. The effect of the online updating is illustrated by an application to predict the number of waiting patients at the emergency department at Aalborg University Hospital.

Keywords: Bayesian networks; conditional linear Gaussian models; incremental learning; fractional updating; fading; real-world application.

1. Introduction

Modern AI-based systems are often dependent on substantial amounts of data for appropriate training of the systems. Bayesian networks (Pearl, 1988; Cowell et al., 1999; Jensen and Nielsen, 2007; Koller and Friedman, 2009; Kjærulff and Madsen, 2013) are quite flexible and may be constructed based on both expert knowledge and historical data. There are methods for learning the qualitative structure of models, e.g., (Spirtes et al., 2000) as well as for estimating the parameters, e.g., (Lauritzen, 1995). This usually results in robust models, yet they may be improved by methods to dynamically adjust the models during their actual use (Spiegelhalter and Lauritzen, 1990; Olesen et al., 1992; Madsen et al., 2017). Such systems are able to adapt parameters to local conditions, e.g., variation in disease patterns at various locations, and to react to temporal changes, for example an increasing inflation or a decreasing frequency of smokers in the population. Online learning can also compensate for imprecise parameters, e.g., conditional probabilities estimated from sparse data, or educated guesses based on expert opinion. The parameters may then be adapted to reflect reality when the system is put in use.

In this paper, we introduce a method for online parameter learning or updating in Bayesian networks with both continuous and discrete variables. In the Conditional Linear Gaussian (CLG) Bayesian network the mean of a continuous variable is a linear function of the continuous parents conditional on the discrete parents and the variance does not depend on the values of the continuous parent. The basic idea of the proposed algorithm is to use the Normal Equation for linear regression to update the mean as new data arrives (as opposed to an iterative Gradient Descent approach or an incremental batch approach that would not update the model between each re-estimation step). The Normal Equation is an analytic approach to finding the coefficients of a linear regression using a least square cost function, see, e.g., Murphy (2022). The trick is that parameter updates can be performed incrementally and online without storing the entire dataset.

Previous work on online learning of Bayesian network parameters includes (Spiegelhalter and Lauritzen, 1990; Olesen et al., 1992; Ratnapinda and Druzdzel, 2015; Madsen et al., 2017). Madsen et al. (2017) and Ratnapinda and Druzdzel (2015) consider different applications of the EM algorithm (Lauritzen, 1995) for parameter learning from a batch of data (referred to as batch EM). Using batch EM, the idea is to collect data in batches and learn parameters off-line, for instance, during maintenance hours as suggested by Ratnapinda and Druzdzel (2015). Adaptive causal probabilistic networks and fractional updating are described in Olesen et al. (1992) who cites Titterington (1976) while adaptive probabilistic networks are described in Russell et al. (1995) and Binder et al. (1997). A similar gradient descent approach is described in Jensen (1999). Madsen et al. (2003) describes how the approach of Olesen et al. (1992) referred to as sequential learning has been implemented in the HUGIN tool. The online EM algorithm of Cappe and Moulines (2009) is another stochastic gradient method for online updating.

The motivation behind this work is the desire to predict patient flows and waiting times for patients at the emergency department of Aalborg University Hospital (UH). Aalborg UH is the largest hospital in the region of North Jutland (one of five regions in Denmark) with 538 beds located in Aalborg (as of July 2020) and 244 beds at other local hospitals in the region. In total the hospital had 6646 full time positions in 2018 (Madsen et al., 2020). The emergency department covers more than half of the activities in the region.

The granularity of the patient flow prediction model by Madsen et al. (2020) is one-hour intervals and it is designed to cope with variations during the day as well as over seasons. Moreover recurring arrangements such as annual music and sport events are incorporated, as well as short term influences as for example weather forecasts. The resulting parameter space is enormous and therefore we have experimented with various ways for online adjustment to improve the precision of the predictions.

In Section 2 we give preliminaries and present the notation used in the paper and in Section 3 we briefly describe the domain of application. Section 4 describes the applied methodology, Section 5 demonstrates the impact of the applied methodology on a motivating example, and Section 6 presents some of the results produced by the developed solution. Finally, Section 7 gives conclusions and pointers to future work.

2. Preliminaries and Notation

A Bayesian network $\mathcal{B} = (G = (V, E), \mathcal{P})$ is a compact way of representing the joint probability distribution over a finite set of discrete variables $\mathcal{X}$. The variables are represented as vertices in the oriented acyclic graph $G = (V, E)$, where the edges E represent direct dependencies between variables represented as nodes V. For each variable $X \in \mathcal{X}, \mathcal{P}$ specifies a conditional probability distribution $P(X \,|\, \mathrm{pa}(X))$ where $\mathrm{pa}(X)$ are the parents of X in G.

$\mathcal{B}$ is a factorization of the joint probability distribution $P(\mathcal{X})$ of $\mathcal{X}$. The joint distribution $P(\mathcal{X})$ decomposes into the product of conditional probability distributions (CPDs) as:

$$P(\mathcal{X}) = \prod_{X \in \mathcal{X}} P(X \,|\, \mathrm{pa}(X))$$

where $\mathcal{X} = \{X_1, \ldots, X_n\}$. From the Bayesian network a junction tree (Jensen et al., 1990) may be constructed. A junction tree exploits the (in)dependencies between variables and stores the joint distribution of $\mathcal{X}$ as a tree where nodes are subsets of $\mathcal{X}$ with associated potentials with an entry for each combination of states in the subset. The junction tree enables efficient computation of the conditional probabilities $P(X \mid \epsilon)$ of variables given evidence ϵ on any combination of other variables. This representation is particularly efficient when G is sparse, but the space requirements remain an inherent problem even for models of moderate size.

The complexity can be reduced by introducing continuous variables. A Conditional Linear Gaussian (CLG) Bayesian network (Lauritzen, 1992; Olesen, 1993; Lauritzen and Jensen, 2001; Kjærulff and Madsen, 2013) extends a Bayesian network with variables with a conditional linear Gaussian distribution for each configuration of discrete parents. The structure of the graph G is restricted to only allow continuous descendants of continuous nodes. We use $\mathcal{X}_\Gamma$ to denote the continuous variables and $\mathcal{X}_\Delta$ to denote the discrete variables such that $\mathcal{X} = \mathcal{X}_\Gamma \cup \mathcal{X}_\Delta$. For each configuration x_I of discrete parents $I \subset \Delta$, a continuous variable Y is specified by a conditional linear Gaussian distribution $\mathcal{N}(\alpha(x_I) + \beta(x_I)^T Z, \sigma(x_I)^2)$, where α is a constant, β is a vector of weights for the continuous parent states represented by the vector Z and β^T is the transpose of β. The variance σ^2 depends only on the configuration x_I of discrete parents. The joint distribution of all variables takes the form of a CG-potential, where continuous variables $\mathcal{X}_\Gamma$ are multivariate normally distributed for each configuration x_Δ of discrete variables $\mathcal{X}_\Delta$:

$$P(\mathcal{X}_\Delta = x_\Delta) * \mathcal{N}_j(\mu(x_\Delta), \sigma^2(x_\Delta)) = \prod_{X \in \mathcal{X}_\Delta} P(X \,|\, \mathrm{pa}(X)) * \prod_{Y \in \mathcal{X}_\Gamma} p(Y \,|\, pa(Y))$$

where $\mathcal{N}_j$ denotes a j-dimensional Gaussian, $j = |\mathcal{X}_\Gamma|$ and p is the density function for Y.

When a system is put in adaptive mode the conditional distributions may be updated based on the actual observed values. In the discrete case conditional probabilities are assumed to be Dirichlet distributed (Spiegelhalter and Lauritzen, 1990). A variable with n states is described by a n-dimensional Dirichlet distribution with parameters $\alpha_1, ..., \alpha_n$. The α's may be interpreted as a contingency table with counts of observed data. The sum of all cases $s = \Sigma_i \alpha_i$ is termed the *equivalent sample size*, and the distribution of X is simply the observed frequencies of the states:

$$P_s(X) = (\alpha_1/s, \alpha_2/s, ..., \alpha_n/s)$$

After a new case is seen the corresponding α is increased by 1 and the frequencies are updated accordingly. If the i'th state is observed the updated distribution is computed as

$$P_{s+1}(X) = (\alpha_1/(s+1), \ldots, (\alpha_i + 1)/(s+1), \ldots, \alpha_n/(s+1))$$

As can be seen it is sufficient to supplement the distribution with a recording of s in order to do the update.

If the parents of a variable are not observed, we may still adjust the distribution of X. This is done by *fractional updating* where a linear combination of Dirichlet distributions is approximated by a single Dirichlet distribution with the correct mean and average variance. We will not go in details with this, but refer the reader to (Spiegelhalter and Lauritzen, 1990) for details.

As the system accumulates experience, the counts will increase, and the dynamics therefore be reduced. This may increase the certainty of static parameters, but in other situations some flexibility is desirable. A more vivid system can be maintained by *fading*, where the equivalent sample size is limited by an upper bound termed *the maximal sample size m.* A fading factor $\lambda = \frac{m-1}{m}$ is then multiplied to the effective sample size.

Our aim is to extend the sketched methods to cope with density functions for continuous variables. Discrete variables are updated as usual, and we seek procedures to update conditional linear Gaussian distributions in the setting of CLG Bayesian networks.

To assess the performance of the prediction models, we use mean absolute error (MAE) and root mean square error (RMSE). MAE is defined as MAE $= \frac{\sum_{i=1}^{N} |y_i - x_i|}{n}$ and RMSE is defined as RMSE $= \sqrt{\frac{\sum_{i=1}^{N} (y_i - x_i)^2}{N}}$ where N is the sample size, and y_i is the true value and x_i is the prediction for case i.

3. Domain of Application

A CLG Bayesian network model for predicting patient flow has been constructed from a combination of domain expert knowledge and historical data using the learning capabilities of HUGIN software (Madsen et al., 2003, 2017) as described by Madsen et al. (2020). The prediction model was estimated off-line and a number of indicator variables are included in the model to adjust for specific (and returning) events such as, e.g., national holidays, vacation, social events and sport events (Madsen et al., 2020). Figure 1 shows a (simplified) excerpt of the model for predicting the number of patients waiting in the emergency department at Aalborg University Hospital. Nodes depicted by double ovals are continuous and the variables *hour, weekday*, and *month* are discrete with their intuitive state spaces. The model is temporal and variables with prefix LAG carries over the values from the previous hour. In order to cope with the complexity of the domain, the number of waiting patients is modelled as a continuous variable even though that it is, of course, discrete by nature.

The domain of application and a software system for predicting patient flow at the emergency department of Aalborg University Hospital has been described in (Madsen et al., 2020). The model in Figure 1 is a simplified version of the model deployed in the software system. It has a total of seven variables where three are discrete and four are conditional linear Gaussian. The model implemented in the software system is a dynamic Bayesian network in order to predict future patient flow beyond one hour. In the simplified model,

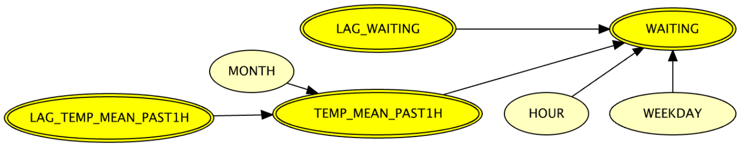

Figure 1: An excerpt of the model for predicting patient flow.

the dynamics of the domain of application is captured by the lag variables that are always observed (we know the number of patients that arrived in the past hour).

Using the linear Gaussian distribution reduces the number of parameters to be estimated. This is important as the dataset available covers a short period of time and there is some dynamics in the underlying population due to, for instance, changes in the geographical area covered by the emergency department at Aalborg University Hospital. The aim of this paper is not to describe the model in detail but rather to present the approach for updating the parameters of the model as data is collected.

4. Methodology

In this section we will describe the procedures to update the conditional Gaussian density functions as new cases are run through the system. A continuous variable is updated for the observed configuration of its discrete parents. In a general setting, no learning takes place for unobserved configurations of discrete variables (in the patient flow model all discrete variables are observed).

Consider a continuous variable Y with discrete parents I in configuration x_I. The distribution of Y conditional on I is assumed to be a CLG distribution where the mean is a linear regression function of the continuous parents $\text{pa}(Y) \cap \mathcal{X}_\Gamma = \{Z_1, \ldots, Z_m\}$ on the form $\alpha(x_I) + \sum_{j=1}^{m} \beta_j(x_I) z_j$ and the variance is $\sigma(x_I)^2$.

To fit a linear regression to a set of cases using a least square cost function, we can, e.g., use an iterative Gradient Descent approach or an analytical approach such as the Normal Equations. We will use the latter as it is a one-step algorithm that gives a closed-form solution to minimize the loss function (least square cost function) and we expect to have a (very) low number of continuous parents of each variable. We describe for the three cases $|\text{pa}(Y) \cap \mathcal{X}_\Gamma| = 0, 1, 2$, which are used in Figure 1, how to update the parameters of the model incrementally without storing the entire dataset. To simplify the notation, we do not index the equations by the configuration x_I of the discrete parents I.

4.1 Basic case

For a continuous variable with no parents the potential is a simple one-dimensional Gaussian $\mathcal{N}(\mu, \sigma^2)$. If the distribution of the variable Y is described by a sample of N observations $y_1, \ldots, y_N$, we estimate the mean μ_N and variance σ_N^2 of Y from the standard formulas

$$\mu_N = \frac{1}{N} \sum_{i=1}^{N} y_i \quad \text{and} \quad \sigma_N^2 = \frac{1}{N-1} \sum_{i=1}^{N} (y_i - \mu_N)^2$$

Rewriting the squared expression

$$(y_i - \mu_N)^2 = \left(\sum_{i=1}^{N} y_i^2\right) - \frac{1}{N}\left(\sum_{i=1}^{N} y_i\right)^2$$

we can calculate the updated mean and variance when a new instance y_{N+1} becomes known:

$$\mu_{N+1} = \frac{1}{N+1}\left(\sum_{i=1}^{N} y_i + y_{N+1}\right) \quad \text{and} \quad \sigma_{N+1}^2 = \frac{1}{N}\left(\left(\sum_{i=1}^{N} y_i^2 + y_{N+1}^2\right) - \frac{1}{N+1}\left(\sum_{i=1}^{N} y_i + y_{N+1}\right)^2\right)$$

It follows that the triple $(N, \sum_{i=1}^{N} y_i, \sum_{i=1}^{N} y_i^2)$ is a sufficient statistic to update the model.

4.2 Variables with one parent

For a continuous node Y with one continuous parent X, we have $y = \beta_0 + \beta_1 x$. We find estimates for μ_Y, μ_X and β_1

$$\mu_Y = \frac{1}{N}\sum y_i \quad \text{and} \quad \mu_X = \frac{1}{N}\sum x_i \quad \text{and} \quad \hat{\beta}_1 = \frac{ss_{xy}}{ss_{xx}}$$

where

$$ss_{xx} = \sum x^2 - \frac{1}{N}\left(\sum x\right)^2 \quad \text{and} \quad ss_{xy} = \sum xy - \frac{1}{N}\left(\sum x \sum y\right)$$

are proportional to the variance of X, respectively, the covariance between X and Y, q_{YX}

$$\begin{aligned}
q_{YX} &= \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \mu_Y)(x_i - \mu_X) \\
&= \frac{1}{N-1}\left(\sum_{i=1}^{N} y_i x_i - \mu_X \sum_{i=1}^{N} y_i - \mu_Y \sum_{i=1}^{N} x_i + \sum_{i=1}^{N} \mu_Y \mu_X\right) \\
&= \frac{1}{N-1}\left(\sum_{i=1}^{N} y_i x_i - \frac{1}{N}\sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i - \frac{1}{N}\sum_{i=1}^{N} y_i \sum_{i=1}^{N} x_i + \frac{1}{N}\sum_{i=1}^{N} y_i \sum_{i=1}^{N} x_i\right) \\
&= \frac{1}{N-1}\left(\sum_{i=1}^{N} y_i x_i - \frac{1}{N}\sum_{i=1}^{N} y_i \sum_{i=1}^{N} x_i\right)
\end{aligned}$$

Finally, the updated values for β_0 is computed as $\hat{\beta}_0 = \mu_Y - \hat{\beta}_1 \mu_X$ and the variance for Y is adjusted to $\sigma_Y^2 - \frac{q_{YX} q_{XY}}{\sigma_X^2}$ as it is conditioned on X being known.

It follows that $(N, \sum x, \sum y, \sum x^2, \sum y^2, \sum xy)$ is a sufficient statistic for Y and we must add $\sum xy$ in addition to what is already stored. The elements of the sufficient statistic can be increased incrementally when a new case arrives similarly to how this is done for the basic case. In the case of missing values, the computed mean is used as the true value in the updating equations.

4.3 Variables with two parents

For a continuous node Y with two continuous parents X_1 and X_2 we have $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$. Without going into details, we simply state the formulas for $\hat{\beta}_i$

$$
\begin{aligned}
\hat{\beta}_0 &= \bar{y} - \hat{\beta}_1 \bar{x}_1 - \hat{\beta}_2 \bar{x}_2 \\
\hat{\beta}_1 &= \frac{\sum x_1 y (\sum x_2)^2 - \sum x_2 y \sum x_1 x_2}{\sum x_1^2 \sum x_2^2 - (\sum x_1 x_2)^2} \\
\hat{\beta}_2 &= \frac{\sum x_2 y (\sum x_1)^2 - \sum x_1 y \sum x_1 x_2}{\sum x_1^2 \sum x_2^2 - (\sum x_1 x_2)^2}
\end{aligned}
$$

and the variance for Y is now adjusted by an additional term to $\sigma_Y^2 - \frac{q_{YX_1} q_{X_1Y}}{\sigma_{X_1}^2} - \frac{q_{YX_2} q_{X_2Y}}{\sigma_{X_2}^2}$.

It follows that $(N, \sum x_1, \sum x_2, \sum y, \sum x_1^2, \sum x_2^2, \sum y^2 \sum x_1 x_2, \sum x_1 y, \sum x_2 y)$ is a sufficient statistic for Y. This generalises nicely, such that an extra term $\sum x_i y$ is needed for each parent X_i to Y and similarly an additional factor $\sum x_i x_j$ has to be added for each combination of parents X_i and X_j.

Again, in this case, the elements of the sufficient statistic can be increased incrementally when a new case arrives similarly to how this is done for the basic case. In the case of missing values, the computed mean is used as the true value in the updating equations.

4.4 Simple fading Mechanism

The proposed fading mechanism is implemented by multiplying each element of the sufficient statistic by the fading factor λ for each variable prior to updating. For instance, in the case of two parents, the element $\sum xy$ is updated as $x_{N+1} y_{N+1} + \lambda * \sum xy$ where x_{N+1} and y_{N+1} are the values of X and Y in the new case $N+1$.

5. Motivating Example

Figure 2 illustrates the performance of the proposed methodology on a motivating example with the objective to predict the variable Waiting that represents the number of waiting patients at a specific point in time. The figure shows the training data (brown), the test data (blue), and predictions made with the model without updating (orange) and different values of λ (green, red, and purple). The x-axis covers a series of five time points that are repeated four times producing a dataset of twenty values. This data is repeated twice. The datasets can be interpreted as reflecting values over four days where each day only has five hours. This means that the training data (when conditioning on HOUR) has five hours with four values each distributed over four days. The test period is similar to the train period where the value five has been added to each value shifting the curve up by five. The aim of the example is to demonstrate that updating as expected over time improves performance by reducing the error in the predictions. We use a simple model M similar to Figure 1 to predict the number of waiting patients (for the example we ignore the fact that predictions cannot be negative).

It is clear from a comparison of the orange plots in the figure that updating improves the performance of the predictions. The MAE for each of the cases where M is not updated and updated, respectively, is MAE(M) = 6.4 and MAE($M^*(\lambda = 1)$) = 5.08 where M^* is M

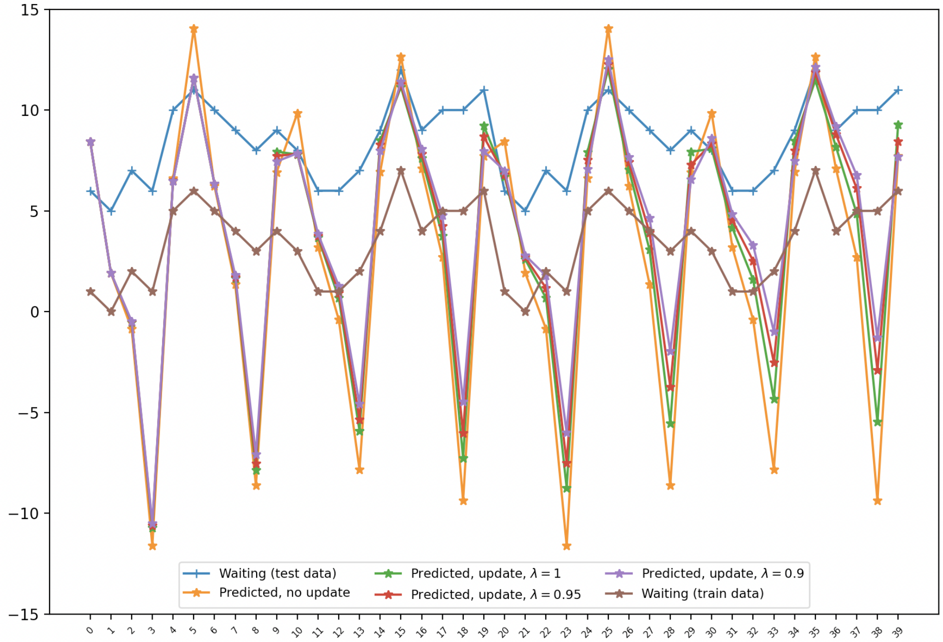

Figure 2: Train, test, and prediction without and with updating (different λ values).

updated. This corresponds to a 21% improvement. The fading has in this case been set to $\lambda = 1$ reflecting that cases are accumulated. With fading factor $\lambda = 0.95$, the performance is $\text{MAE}(M^*(\lambda = 0.95)) = 4.77$ while the performance is $\text{MAE}(M^*(\lambda = 0.9)) = 4.53$ for $\lambda = 0.9$. In this example, the fading mechanism improves performance.

6. Experimental Results

In this section, we describe the results of an experimental analysis performed on anonymised historical data from the emergency department at Aalborg University Hospital.

6.1 Setup

The experimental analysis uses the simplified prediction model shown in Figure 1. The patient flow at the emergency department correlates with the weather. The model uses observational data on mean temperature the past 1 hour. This data is accessed through Danish Meteorological Institute's (DMI) Open Data Application Programming Interface (API). DMI Open API provides free and open access to DMI's data[1][2] through a REST API.

The dataset used in the experiments covers the period from March 2019 until end of 2020. There was a change in the annotation of data in March, 2019, which means that data prior to March, 2019 is not available for the experimental analysis.

[1] https://confluence.govcloud.dk/display/FDAPI/Danish+Meteorological+Institute+-+Open+Data

[2] https://www.dmi.dk/frie-data

The analysis uses different subsets of the data from 2019 to estimate the initial values of the parameters of the model. Four different subsets are considered (March to December, July to December, October to December, and December). The initial estimation of the parameters of the model also produces the set of sufficient statistics necessary for the online updating method. The aim is to assess the potential impact of the amount of data on the performance of the updating algorithm.

The experimental analysis is performed with and without updating the parameter values of the model as part of processing the test data. For the experiment with updating the parameter values, we consider ten different values of the fading factor λ. The values are $0.9, 0.91, \ldots, 0.98, 0.99$ and 1 reflecting equivalent sample sizes of $10, 11.1, \ldots, 50, 100$, and accumulation of data, respectively. Furthermore, in the experiments, we set all predicted negative values to zero as it does not make any sense to predict a negative number of waiting patients.

The model is predicting the number of waiting patients classified as orthopaedic patients where the input values are point in time of the prediction (month, weekday and hour), mean temperature past one hour, and the number of waiting patients the previous hour. Please notice that we are introducing a general updating scheme that does not require all discrete variables to be observed.

The methodology is implemented in Python using the HUGIN Python API[3] (Andersen et al., 1989; Madsen et al., 2005). Experiments are performed on an Apple MacBook Pro (M1, 2020, 8 GB RAM) running Monterey.

6.2 Results

The results of the empirical analysis on MAE are shown in Figure 3 (the results for RMSE shows a similar plot that is left out to meet the page restrictions). The figure shows eight curves with four curves in black and four curves in different colors where the black curves represent the MAE values with no updating for the four datasets. Here, the largest training dataset (March to December) produces the smallest MAE and the smallest training dataset (December) produces the largest MAE. The black curves are included for easy reference and they are constant across the x-axis as the MAE in the case of no updating does not depend on the value of λ. The color curves show the MAE for the four different datasets as a function of λ values in the range from 0.9 to 1.

It is clear from the results presented in Figure 3 that accuracy of the predictions improves with the amount of data used to estimate the initial values of the model parameters. The four black curves in the figure shows this. The bottommost black line represents the training dataset March - December with MAE = 0.821, the second bottommost black line represents the training dataset July - December with MAE = 0.853, the second topmost black line represents the training dataset October - December with MAE = 0.941, and the topmost black line represents the training dataset December with MAE = 1.56. Hence, with no updating the performance decreases from MAE = 0.821 to MAE = 1.56 as the number of cases is reduced from covering March - December to only December.

Similarly, considering the case of data accumulation, i.e., $\lambda = 1$, the performance in terms of MAE improves from MAE = 1.08 to MAE = 0.801 as the size of the training dataset

[3] `https://www.hugin.com`

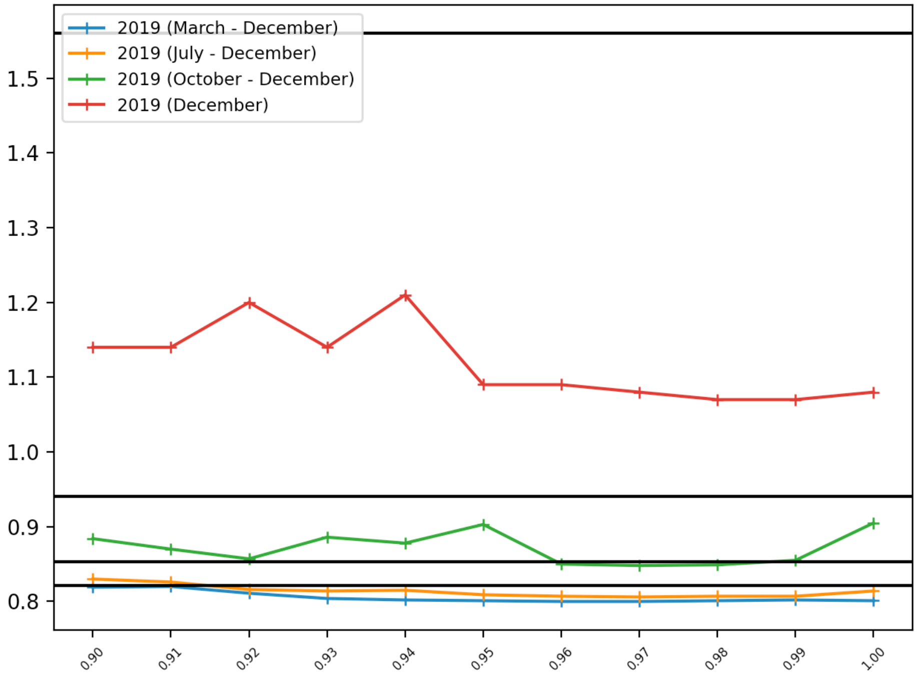

Figure 3: Mean absolute error against λ values for the four different training datasets.

increases. The performance is in all cases except one better than the results obtained using no updating. In fact, all experiments with updating using different values of λ and different training datasets are better than the corresponding experiment with no updating (except a few cases). Reducing the size of the dataset used for the initial parameter estimation has a larger impact when there is no updating than when parameters are updated.

Notice that discrete variables in the model (see Figure 1) are always observed in the experimental analysis as they represent the point in time for which the prediction is made. Hence, the learning is only relevant for the continuous variables in the model. In some situations, data may be missing on the continuous variables for different reasons. In such cases, we use the estimated values as true values.

For the datasets considered in the experiments, the impact of using a fading factor appears to be marginal. In a few cases, the performance is improved over the case of no fading. This is the case for the training set October - December where the performance appears to degrade as λ approaches 1 (this is the green curve in Figure 3). Additional experiments with several more datasets are required to make qualified recommendations on the most appropriate value of λ.

7. Conclusion and Future Work

We have presented a method for online updating of conditional distributions in Bayesian network models with both discrete and continuous variables. The proposed method extends known procedures for online updating of parameters in Bayesian networks with discrete variables to cope with continuous variables that have a conditional linear Gaussian distribution.

The method includes an option to use a fading mechanism to reduce the impact of past data and to avoid the system becoming too insensitive to future cases.

The method is motivated by work on predicting the number of waiting patients at the emergency department of Aalborg University Hospital. The paper includes the results of an experimental analysis where different subsets of 2019 data is used to predict the number of waiting patients in 2020 under different values of the fading factor. The option of online updating is important as the dataset available to estimate the parameters of the prediction model is rather limited. In addition, there has been changes in the geographical area covered by the emergency department of Aalborg University Hospital, which change the patient flow.

The results of the experimental analysis demonstrate that the online updating of conditional linear Gaussian Bayesian networks can improve performance of the predictions.

Acknowledgments

The project is partly financed by regional funds of the European Union through Danish Life Science Cluster's Sundhedsteknologiske Service Program. We thank the reviewers for insightful comments that have helped improved the quality of the paper.

References

S. K. Andersen, K. G. Olesen, F. V. Jensen, and F. Jensen. HUGIN — a Shell for Building Bayesian Belief Universes for Expert Systems. In *Proc. of IJCAI*, pages 1080–1085, 1989.

J. Binder, D. Koller, S. Russell, and K. Kanazawa. Adaptive Probabilistic Networks with Hidden Variables. *Machine Learning*, 29(2):213–244, 1997.

O. Cappe and E. Moulines. Online EM Algorithm for Latent Data Models. *J. Royal Statistical Society Series B (Statistical Methodology)*, 71(3):593–613, 2009.

R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic Networks and Expert Systems.* Springer, 1999.

F. V. Jensen. Gradient Descent Training of Bayesian Networks. In *Proc. ECSQARU*, pages 190–200, 1999.

F. V. Jensen and T. D. Nielsen. *Bayesian Networks and Decision Graphs.* Springer, 2nd edition, 2007.

F. V. Jensen, S. L. Lauritzen, and K. G. Olesen. Bayesian updating in causal probabilistic networks by local computations. *Computational Statistics Quarterly*, 4:269–282, 1990.

U. B. Kjærulff and A. L. Madsen. *Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis.* Springer, 2nd edition, 2013.

D. Koller and N. Friedman. *Probabilistic Graphical Models — Principles and Techniques.* MIT Press, 2009.

S. L. Lauritzen. Propagation of probabilities, means and variances in mixed graphical association models. *J. American Statistical Association (Theory and Methods)*, 87(420): 1098–1108, 1992.

S. L. Lauritzen. The EM algorithm for graphical association models with missing data. *Computational Statistics & Analysis*, 19:191–201, 1995.

S. L. Lauritzen and F. Jensen. Stable local computation with mixed Gaussian distributions. *Statistics and Computing*, 11(2):191–203, 2001.

A. L. Madsen, M. Lang, U. B. Kjærulff, and F. Jensen. The Hugin Tool for Learning Bayesian Networks. In *Proc. of ECSQARU*, pages 594–605, 2003.

A. L. Madsen, F. Jensen, U. B. Kjærulff, and M. Lang. HUGIN - The Tool for Bayesian Networks and Influence Diagrams. *International Journal on Artificial Intelligence Tools 14*, 3:507–543, 2005.

A. L. Madsen, N. S. Jeppesen, F. Jensen, M. S. Sayed, U. Moser, L. Neto, J. Reis, and N. Lohse. Parameter learning algorithms for continuous model improvement using operational data. In *Proc. of ECSQARU*, pages 115–124, 2017.

A. L. Madsen, K. G. Olesen, J. M. Møller, N. Søndberg-Jeppesen, F. Jensen, T. M. Larsen, P. Henriksen, M. Lindblad, and T. S. Christensen. A Software System for Predicting Patient Flow at the Emergency Department of Aalborg University Hospital. In *Proc. of PGM*, pages 617–620, 2020.

K. P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. URL `probml.ai`.

K. G. Olesen. Causal probabilistic networks with both discrete and continuous variables. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(3):275–279, 1993.

K. G. Olesen, S. L. Lauritzen, and F. V. Jensen. aHUGIN: A System Creating Adaptive Causal Probabilistic Networks. In *Proc. of UAI*, pages 223–229, 1992.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Series in Representation and Reasoning. Morgan Kaufmann, 1988.

P. Ratnapinda and M. J. Druzdzel. Learning discrete Bayesian network parameters from continuous data streams: What is the best strategy? *J. Applied Logic*, 13:628–642, 2015.

S. Russell, J. Binder, D. Koller, and K. Kanazawa. Local Learning in Probabilistic Networks With Hidden Variables. In *Proc. of IJCAI*, pages 1146–1152, 1995.

D. Spiegelhalter and S. L. Lauritzen. Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20:579–605, 1990.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. Adaptive Computation and Machine Learning. MIT Press, second edition, 2000.

D. M. Titterington. Updating a diagnostic system using unconfirmed cases. *Applied Statistics*, 25:238–47, 1976.