Fast approximate inference in hybrid Bayesian networks using dynamic discretisation

Helge Langseth[1], David Marquez[2], and Martin Neil[2]

[1] Department of Computer and Information Science,
The Norwegian University of Science and Technology, Norway
[2] School of Electronic Engineering and Computer Science,
Queen Mary University of London, London E1 4NS, United Kingdom

Abstract. We consider inference in a Bayesian network that can consist of a mix of discrete and continuos variables. It is well known that this is a task that cannot be solved in general using a standard inference algorithms based on the junction-tree. A common solution to this problem is to discretise the continuous variables to obtain a fully discrete model, in which standard inference can be performed. The most efficient discretisation procedure in terms of cost of inference is known as *dynamic discretisation*, and was published by Kozlov and Koller in the late 90's. In this paper we discuss an already published simplification to that algorithm by Neil et al. The simplification is in practise orders of magnitudes faster than Kozlov and Koller's technique, but potentially at the cost of some lack of precision. We consider the mathematical properties of Neil et al.'s algorithm, and challenge it by constructing models that are particularly difficult for that method. Some simple modifications to the core algorithm are proposed, and the empirical results are very promising, indicating that the simplified procedure is feasible also for very challenging problems.

1 Introduction

By a *hybrid* Bayesian network we denote a network where some variables are discrete, others are continuous. It is well known that the exact inference schemes currently employed (e.g., [1]) only work for some particular classes of hybrid BNs. The most common strategies for performing inference in a hybrid BN can roughly be divided into three categories: Firstly, a subset of models (commonly referred to conditional Gaussian models) allow exact inference.Secondly, approximate inference procedures like stochastic sampling can be employed. Finally, one can make explicit changes in the representations of the conditional distribution functions defined for each variable in order to facilitate exact inference. The most prominent among these alternations is *discretisation*, i.e., to "translate" all continuous variables into discrete ones [2]. During discretisation, each new discrete variable has to be given an adequate number of states to capture the associated continuous variable sufficiently well, and the tradeoff between model precision and model complexity is therefore particularly evident during this task.

Consider a continuous variable with k continuous parents, and assume we discretise each variable into m states. Doing so, we create conditional probability tables with a total size of order $\mathcal{O}(m^k)$ to represent the discretised model. This makes a fine-grained discretisation computationally ineffective, even for moderate values of k. Thus, traditional discretisation heuristics like "equal-mass" and "equal-width" discretisation have been replaced by techniques that use *non-uniform* discretisation. Here, a finer discretisation is employed were it pays the most, an idea pioneered by Kozlov and Koller [3], who aimed at finding a discretised probability density function (pdf) as close to the original pdf as possible in terms of the KL distance [4]. In the following we consider a pdf over a multivariate vector $\boldsymbol{X}$. We assume that $\boldsymbol{X}$ is continuous, as it is trivial to extend our discussion to hybrid domains. Let the pdf $f(\boldsymbol{x})$ be defined on $\boldsymbol{x} \in \boldsymbol{\Omega} \subseteq \mathbb{R}^d$. $\boldsymbol{\Omega}$ is partitioned into hypercubes (or "subsets") $\boldsymbol{\omega}_k$, $k = 1, \ldots, t$ such that $\boldsymbol{\omega}_i \cap \boldsymbol{\omega}_j = \emptyset$ and $\cup_{k=1}^t \boldsymbol{\omega}_k = \boldsymbol{\Omega}$. A discretisation $\bar{f}(\boldsymbol{x})$ on $f(\boldsymbol{x})$ wrt. $\{\boldsymbol{\omega}_k\}_{k=1}^t$ is a non-negative function of $\boldsymbol{x} \in \boldsymbol{\Omega}$, constant on each hypercube $\boldsymbol{\omega}_\ell$ (i.e., $\bar{f}(\boldsymbol{x}) = \bar{f}_\ell$ for constant $\bar{f}_\ell$ when $\boldsymbol{x} \in \boldsymbol{\omega}_\ell$), and which integrates to unity (so $\sum_{k=1}^t \bar{f}_k \cdot |\boldsymbol{\omega}_k| = 1$, where we use $|\boldsymbol{\omega}_\ell| = \int_{\boldsymbol{x} \in \boldsymbol{\omega}_\ell} d\boldsymbol{x}$ to denote the volume of the hypercube $\boldsymbol{\omega}_\ell$). The KL distance from f to $\bar{f}$ can be calculated by summing over the partitions of the discretisation [4, 3], giving

$$D\left(f \,\|\, \bar{f}\right) = \sum_{k=1}^{t} \int_{\boldsymbol{x} \in \boldsymbol{\omega}_k} f(\boldsymbol{x}) \log\left(\frac{f(\boldsymbol{x})}{\bar{f}_k}\right) \mathrm{d}\boldsymbol{x}. \tag{1}$$

It is easy to verify that given the subsets $\{\boldsymbol{\omega}_k\}_{k=1}^t$, the discretised pdf closest to $f(\boldsymbol{x})$ in KL distance is found by choosing $\bar{f}_\ell = \int_{x \in \boldsymbol{\omega}_\ell} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} / |\boldsymbol{\omega}_\ell|$, see [3]. Finding a "good" discretisation of $f(\boldsymbol{x})$ for $\boldsymbol{x} \in \boldsymbol{\Omega}$ therefore amounts to determining the set of hypercubes $\boldsymbol{\omega}_k$ as defined above.

Define further the notation that $f_\ell^\uparrow = \max_{\boldsymbol{x} \in \boldsymbol{\omega}_\ell} f(\boldsymbol{x})$ and $f_\ell^\downarrow = \min_{\boldsymbol{x} \in \boldsymbol{\omega}_\ell} f(\boldsymbol{x})$. Now, Kozlov and Koller [3] showed that the contribution from each term in the sum of Equation (1) can be bounded above by

$$\begin{aligned} &\int_{\boldsymbol{x} \in \boldsymbol{\omega}_\ell} f(\boldsymbol{x}) \log\left(\frac{f(\boldsymbol{x})}{\bar{f}_\ell}\right) \mathrm{d}\boldsymbol{x} \leq \\ &\qquad \left[\frac{f_\ell^\uparrow - \bar{f}_\ell}{f_\ell^\uparrow - f_\ell^\downarrow} f_\ell^\downarrow \log\left(\frac{f_\ell^\downarrow}{\bar{f}_\ell}\right) + \frac{\bar{f}_\ell - f_\ell^\downarrow}{f_\ell^\uparrow - f_\ell^\downarrow} f_\ell^\uparrow \log\left(\frac{f_\ell^\uparrow}{\bar{f}_\ell}\right)\right] |\boldsymbol{\omega}_\ell| \,. \end{aligned} \tag{2}$$

This motivates a greedy discretisation strategy, where the following two steps are repeated until some termination criteria is met: *i*) Calculate the upper-bound of the contribution to the total KL distance on each hypercube $\boldsymbol{\omega}_k$, $k = 1, \ldots, t$, using Equation (2), and *ii*) Partition the hypercube with the highest bound into two parts.

Kozlov and Koller [3] reported that while this strategy works well for Bayesian networks without evidence inserted, it can become quite poor when posterior probabilities are calculated from (low-probability) evidence. This motivates *dynamic* discretisation, where the discretisation process is conducted while *taking*

```
1  Function dynDiscMarginal
   Input  : BN $\mathcal{B}$, Evidence $\boldsymbol{\epsilon}$, Query $\mathcal{Q}$.
   Output: Approximation of the posterior distribution $f(q|\boldsymbol{\epsilon})$ in $\mathcal{B}$.
2  $\Omega \leftarrow$ Initial discretisation of $\boldsymbol{X}$;
3  $\boldsymbol{Y} \leftarrow \boldsymbol{X} \setminus \boldsymbol{\epsilon}$;
4  repeat
5      $\mathcal{D} \leftarrow$ Discretised version of $\mathcal{B}$ using $\Omega$ ;
6      Calculate $P_{\mathcal{D}}(\boldsymbol{Y}|\boldsymbol{\epsilon})$;
7      foreach variable in $\boldsymbol{Y}$ do
8          $\Omega_i \leftarrow$ sort($\Omega_i$, bestSplit($\Omega_i, P_{\mathcal{D}}(Y_i|\boldsymbol{\epsilon})$));
9      end
10 until converged;
11 return $P_{\mathcal{D}}(Q|\boldsymbol{\epsilon})$;
```

Algorithm 1: Skeleton for the *dynamic discretisation*-algorithm.

evidence into account. The discretisation is an integral part of the inference algorithm, and the discretisation is done at the level of the cliques, thus ensuring a close-to-optimal discretisation of the domain as a whole. Unfortunately, though, this algorithm has computational issues, which have prevented it from being commonly used in practice.[1] For instance, the algorithm requires re-implementation of the message passing algorithm for inference (communicating objects called "weights", which are used to re-adjust the discretisation when evidence is found in low-probability regions of the density together with the standard messages), it uses specialised data structures called binary split partition trees for which the standard inference operations must be defined, and it must find or approximate the minimum and maximum values of potentially high-dimensional functions on each hypercube to utilise the bound in Equation (2).

Neil et al. [5] proposed a refinement of Kozlov and Koller's work, defining an algorithm which operates using only the *standard inference engine for discrete variables.* The key idea of the algorithm is to discretise each variable *separately* by utilising Equation (2) to discretise each variable based on that variable' approximated *posterior marginal distribution.* This results in an inference procedure, which is extremely fast, also for BN models of considerable size.[2] The main steps are given in Algorithm 1: The algorithm takes a BN $\mathcal{B}$ over variables $\boldsymbol{X}$, evidence $\boldsymbol{\epsilon}$ and a query Q as input, and starts by roughly discretising all variables $\boldsymbol{X}$. Evidence variables are discretised by defining split-points just below and just above their observed values; these discretisations will not be refined later. Unobserved variables, called $\boldsymbol{Y}$ for ease of reference, are initially discretised by dividing their support into a predefined number of intervals. The discretised version of the

[1] To the best of our knowledge, there is no implementation of Kozlov and Koller's algorithm publicly available.

[2] The dynamic discretisation algorithm [5] is implemented in the AgenaRisk software package. A free version of AgenaRisk can be downloaded from `http://www.agenarisk.com/`.

hybrid BN $\mathcal{B}$ is denoted $\mathcal{D}$, and we can use a standard inference engine for discrete Bayesian networks to calculate any posterior distribution in $\mathcal{D}$; $P_{\mathcal{D}}(\boldsymbol{Y}|\boldsymbol{\epsilon})$ is used to explicitly denote that $P(\boldsymbol{Y}|\boldsymbol{\epsilon})$ is calculated in $\mathcal{D}$. The discretisation for variable Y_i (denoted by Ω_i) is refined by repeating the following steps: Firstly, the posterior distribution over $\boldsymbol{Y}$ is calculated in $\mathcal{D}$ (variables defined using the current discretisation). Secondly, in line 8, the discretisation Ω_i for Y_i is refined by adding a new split-point at the mid-point of the current interval with the highest KL bound (calculated using Equation (2)) if this is deemed beneficial; the actual split-point is found by the external function `bestSplit`. The iteration is terminated as soon as a convergence measure is met. In the following we will examine Algorithm 1 in more detail, and discuss the major differences between that algorithm and Kozlov and Koller's work [3].

2 Discretising a single variable

The first important difference between [5] and [3] is that Algorithm 1 uses the *marginal discretised pdf* when calculating $f_\ell^\downarrow$, $f_\ell^\uparrow$ and $\bar{f}_\ell$ which later are used to find the best interval to split (Line 8). Kozlov and Koller, on the other hand, use the (potentially multivariate) continuous functions. Algorithm 1 thus considers the discretised pdf as a step-function (see Fig. 1), where the minimum and maximum values are easily established: $f_\ell^\downarrow$ for an interval ω_ℓ is simply defined as the minimum value obtained by the discretised pdf on ω_ℓ and its two neighbouring intervals, and $f_\ell^\uparrow$ is found similarly. $\bar{f}_\ell$ is defined as the pdf value at ω_ℓ, although special rules are employed if ω_ℓ holds a mode of the discretised pdf, in which case $\bar{f}_\ell$ is defined as the average value over the three intervals. We note that this approach typically will over-estimate the KL bound in Equation (2), and in particular when the derivative of the true pdf is large in absolute value, thus leading to slightly higher discretisation effort than optimal in those areas. Indications of this effect can be seen in Fig. 2, where Part (a) shows the results of discretising the standard Normal. The optimal discretisation found by simulated annealing is shown with a solid line, and the results of Algorithm 1 are shown with a dashed line. The results are not that different in Part (a), which is based on 10 split-points. However, Part (b) shows the discontinuity points for a discretisation using 24 split-points; the points chosen by Algorithm 1 are in the upper row (drawn as circles) and the approximate optimal solution found by simulated annealing is shown in the lower row (crosses). It is evident that Algorithm 1 puts less effort than optimal at the tales and close to the mode of the pdf, and makes the discretisation around ± 1 (where f is changing the fastest) finer than required.

Finally, Fig. 3 shows the KL divergence from a standard Normal distribution to the discretised version found by Algorithm 1 (solid line) and [3] (dashed line). The number of intervals used during discretisation is given on the x-axis. For comparison, we also report the results found by simulated annealing (dotted line).

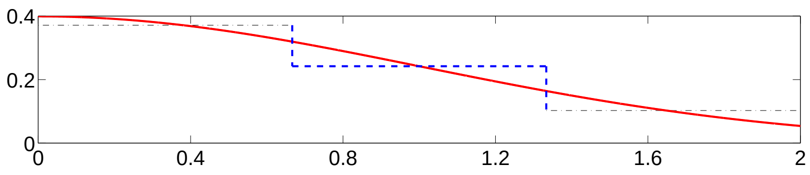


Fig. 1. Each interval is characterised by its discretised pdf.

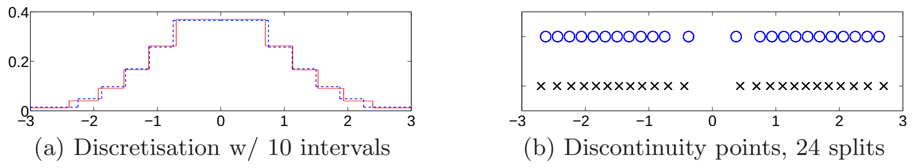


Fig. 2. Part (a): The discretised pdf after (approximately) optimal discretisation (solid line) and the results of Algorithm 1 (dashed line). Part (b): The discontinuity-points chosen by Algorithm 1 (upper row, circles) and the approximately optimal solution found by simulated annealing (lower row, crosses).

3 Multivariate distributions

Define the conditional KL divergence from one conditional distribution $f(\boldsymbol{y}|\boldsymbol{x})$ to another $\tilde{f}(\boldsymbol{y}|\boldsymbol{x})$ to be

$$D\left(f(\boldsymbol{y}|\boldsymbol{x}) \parallel \tilde{f}(\boldsymbol{y}|\boldsymbol{x}) \right) = \int_{\boldsymbol{x}} f(\boldsymbol{x}) \int_{\boldsymbol{y}} f(\boldsymbol{y}|\boldsymbol{x}) \log \frac{f(\boldsymbol{y}|\boldsymbol{x})}{\tilde{f}(\boldsymbol{y}|\boldsymbol{x})}\, d\boldsymbol{y}\, d\boldsymbol{x},$$

so that we can calculate the KL distance between two joint distributions as

$$D\left(f(\boldsymbol{x},\boldsymbol{y}) \parallel \tilde{f}(\boldsymbol{x},\boldsymbol{y}) \right) = D\left(f(\boldsymbol{x}) \parallel \tilde{f}(\boldsymbol{x}) \right) + D\left(f(\boldsymbol{y}|\boldsymbol{x}) \parallel \tilde{f}(\boldsymbol{y}|\boldsymbol{x}) \right).$$

Let us look at the behaviour of Algorithm 1 when we, for simplicity of exposition, consider the case where $\boldsymbol{\epsilon} = \emptyset$ (extending the results to the general case is straight forward), and define $\mathrm{pa}\,(z)$ to denote the parents of Z in the Bayesian network. Now it is easy to verify that the optimal discretisation $\bar{f}$ minimises

$$D\left(f(\boldsymbol{y}) \parallel \bar{f}(\boldsymbol{y}) \right) = \sum_i D\left(f(y_i|\mathrm{pa}\,(y_i)) \parallel \bar{f}(y_i|\mathrm{pa}\,(y_i)) \right),$$

Kozlov and Koller [3] obtains this by looking at the posterior joints at the clique level during their discretisation process. On the other hand, Algorithm 1 is looking at $\sum_i D\left(f(y_i) \parallel \bar{f}(y_i) \right)$, i.e., minimises KL-distances between *marginal distributions*, and thereby partly disregards the effect of the correlations between the random variables during discretisation, which can potentially result in an

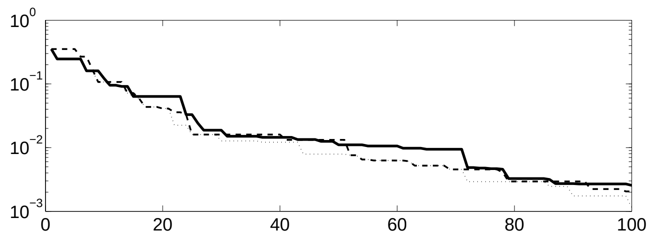


Fig. 3. The KL distance from the true pdf to the discretised pdf as a function of the number of intervals. The results of the "optimal" discretisation (found by simulated annealing) is given with a thin dotted line, and shown together with the results obtained using the true (dashed line) and approximated pdf (solid line) to find $f^{\downarrow}$and $f_{\ell}^{\uparrow}$ in the bound (Equation (2)). Note the log-scale on the y-axis.

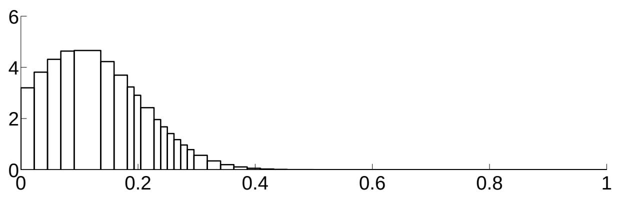


Fig. 4. Conditional distribution for $X|\{Y = .1\}$.

inferior discretisation. To investigate this further, we will now stress-test Algorithm 1 using a simple network consisting of only two nodes, $X \to Y$, but where the distributions are designed to make the models difficult for Algorithm 1.

3.1 Conditioning on a uniformly distributed variable

Consider a Bayesian network as described above, let X follow a uniform distribution on $[0, 1]$ and let $Y|\{X = x\} \sim N(\mu = x, \sigma^2 = .1^2)$. Equation (2) will determine that there is no benefit from discretising X beyond the initial discretisation performed in Line 2 of Algorithm 1 (the pdf will remain at $\bar{f}(x) = 1$ independently of how many intervals the domain of X is discretised into). We can easily solve this by simply insisting on always refining the discretisation of every variable as the algorithm moves along, and use a heuristic like "*split the interval with the highest probability mass through its centre*" to guide our discretisation of X. The results in Fig. 4 have been obtained following this simple heuristic, and we present the conditional distribution of $X|\{Y = .1\}$. The correct result is a truncated Gaussian with its mode at 0.1, which is in good correspondence with the obtained results. The vanilla version of the algorithm fails to represent the correlation between X and Y.

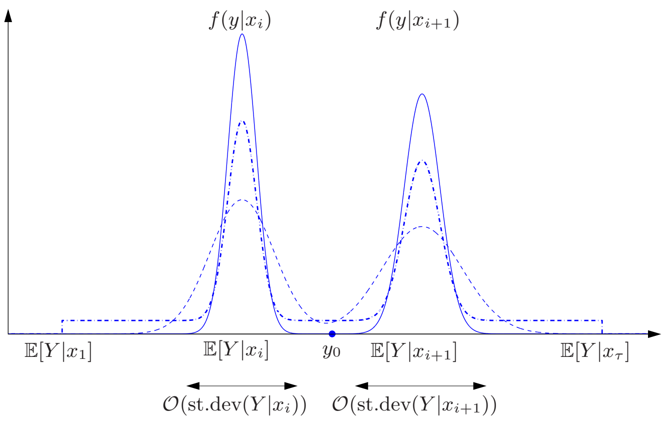


Fig. 5. The original conditional distributions $f(y|x)$ is evaluated for the two points x_i and x_{i+1}; shown with a solid line. Tempered versions of the conditional pdfs are shown with a dashed line, and the uniform fill-in approximation with dash-dots.

3.2 The resolution problem

We now move to models that are "almost deterministic", meaning that the conditional variance $Y|\{X = x\}$ is small compared to the variance of X. We say that these models suffer from the *resolution problem*[3], and the difficulties occur as one tries to populate the conditional probability tables of the discretised model, i.e., while defining $P_{\mathcal{D}}(Y|X)$ in the discretised model $\mathcal{D}$.

Let us start by looking at why this problem arrises, before discussing how it can be solved. Consider a variable Z with mean μ_Z and standard deviation σ_Z, where we by Chebyshev's inequality have for any $\kappa > 0$ that

$$P\left(|Z - \mu_Z| \geq \kappa \cdot \sigma_Z\right) \leq \frac{1}{\kappa^2}. \tag{3}$$

Assume now that we are looking to calculate the conditional probabilities that are required to define the discretised model. For the intervals $\omega_X = [\alpha, \beta]$ and ω_Y (defined for X and Y, respectively), we thus need to calculate $P(Y \in \omega_Y|x \in \omega_X)$. By simple manipulation we get

$$P(Y \in \omega_Y|x \in \omega_X) \propto \int_{y\in\omega_Y} \int_{x\in[\alpha,\beta]} f(y|x)\, dy\; f(x)\, dx,$$

which means that we for each $y_0 \in \omega_Y$ should calculate $\int_{x\in[\alpha,\beta]} f(y_0|x)\, f(x)\, dx$.

The integration will in general have to be done numerically, meaning that we will define a level of granularity, τ, an evaluation set containing τ ordered points

[3] The problem was raised in the context of Algorithm 1 by Roger F. Sewell at Cambridge Consultants.

$\{x_1, \ldots, x_\tau\}$ for which $\alpha = x_1 < x_2 < x_3 < \ldots < x_\tau = \beta$, and a set of constants (or "weights") $\{w_1, \ldots, w_\tau\}$. We then use the approximation

$$\int_{x\in[\alpha,\beta]} f(y_0|x)\, f(x)\, dx \approx \sum_{i=1}^{\tau} w_i \cdot f(y_0|x_i)\, f(x_i). \tag{4}$$

Here the values for w_i are chosen depending on which numerical integration scheme is employed (giving rise to composite versions of, e.g., the *rectangle rule*, the *trapezoid rule*, or *Milne's rule* for numerical integration). Consider Fig. 5, where the conditional pds $f(y|x)$ are shown for $x = x_i$ and $x = x_{i+1}$ (solid line). The length of the area around $f(y|x_j)$ which contributes to the integral is of the order of the standard deviation of $Y|\{X = x_j\}$ (see Equation (3)). Notice that neither $f(y|x_i)$ nor $f(y|x_{i+1})$ contribute to the evaluation of the integral in Equation (4) for the choice of y_0 in this example. To evaluate Equation (4) with desired precision, a rule-of-thumb is therefore to choose τ such that

$$\tau \gg \frac{|\mathbb{E}[Y|X=\alpha] - \mathbb{E}[Y|X=\beta]|}{\min\{\text{st.dev}(Y|X=\alpha), \text{st.dev}(Y|X=\beta)\}}. \tag{5}$$

On the other hand, large values for τ can make the numerical integration extremely slow, and render the approach unsuitable for practical applications. This will be the case for models, which are affected by the resolution problem.

Our solution to performing the numerical integration in Equation (4) with sufficient precision using a limited calculation effort is inspired by *tempering* [6], which was developed to enable Markov chain Monte Carlo samplers to operate efficiently when confronted with multi-modal distributions. Let $\tilde{f}(y|\boldsymbol{x}, T)$ be the tempered version of the conditional pdf $f(y|\boldsymbol{x})$ at "temperature" T. Formally we define $\tilde{f}(y|\boldsymbol{x}, T) = \frac{1}{Z_T} \exp\left(\log\left[f(y|\boldsymbol{x})\right]/T\right)$, where Z_T is a function of T chosen to ensure that $\tilde{f}(y|\boldsymbol{x}, T)$ is a density. The tempered distribution is identical to $f(y|\boldsymbol{x})$ for $T = 1$, it approaches a uniform as T grows towards infinity, and can be seen as a smoothed version of $f(y|\boldsymbol{x})$ for T-values in-between. As an example we mention that the tempered version of a Gaussian with variance σ^2 at temperature T is a Gaussian with variance $T \cdot \sigma^2$. Fig. 5 includes tempered versions of $f(y|x_i)$ and $f(y|x_{i+1})$ drawn with a dashed line. As the tempered pdfs are "smeared out" versions of the true pdfs, they are easier to integrate (require a smaller granularity) than the originals. We note that it is not crucial to have an exact representation of the true pdfs during the first iterations of the discretisation procedure, when the goal is to roughly find which parts of a domain that contains probability mass. It is later, as the discretisation gets more refined, an accurate description of the true pdfs is required. However, at that time the problems with the numerical integration are also less prominent, because the discretisation has already reached an acceptable quality. Our plan is therefore to start the discretisation using a rather high temperature, and gradually cool the temperature towards unity as the discretisation gets more fine-grained.

Unfortunately, tempering can be time-demanding in situations where the tempering distribution is not available in analytical form. We therefore approximate the tempering process by what we call a "*uniform fill-in*". The uniform fill-in of the conditional distribution $f(y|x)$ for $x \in [\alpha, \beta]$ is simply a mixture between the original distribution $f(y|x)$ and a uniform distribution on the interval $\big[\mathbb{E}[Y|X=\alpha], \mathbb{E}[Y|X=\beta]\big]$. The result of using a uniform fill-in is shown in Fig. 5 (dash-dotted line). It resembles tempering in that it "widens" the support of the integrand in Equation (4), thus initially allows the numerical integration procedure to operate with a lower granularity. As the discretisation is refined, less weight is put on the uniform, thereby enabling a faithful representation of the true distribution as the discretisation process approaches its end. This is summarised in Algorithm 2, which takes the integration problem and the mixture probability as inputs. The filled-in distribution is defined as $\tilde{f}(y|x)$ in Line 2, where $U(y; a, b)$ is used to denote the pdf of the uniform distribution for Y on $[a, b]$. τ is then calculated according to the heuristic in Equation (5), but bounded above by a constant $\tau_{\max}$; note that it is $\tilde{Y}$ with pdf $\tilde{f}(y|x)$, which is used to find τ. The evaluation set and the weights are defined in Line 4; here giving a uniform spread and the trapezoidal rule, but other definitions are possible. Finally, the results are calculated in Line 5 using Equation (4).

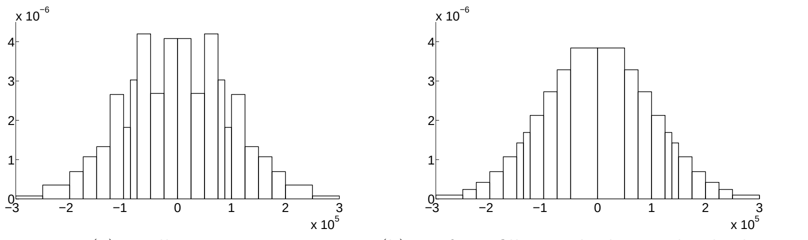
```
1 Function numericalIntegration
  Input  : Interval $[\alpha, \beta]$, pdfs $f(x)$ and $f(y|x)$, $y_0 \in \omega_Y$, mixture $p$.
  Output: Numerical approximation of $\int_{x\in[\alpha,\beta]} f(x) f(y_0|x)\,dx$.
2 $\tilde{f}(y|x) \leftarrow p \cdot U(y; \mathbb{E}[Y|X=\alpha], \mathbb{E}[Y|X=\beta]) + (1-p)\cdot f(y|x)$;
3 $\tau \leftarrow \max\left\{\tau_{\max}, \frac{\left|\mathbb{E}[\tilde{Y}|X=\alpha] - \mathbb{E}[\tilde{Y}|X=\beta]\right|}{\min\left\{\text{st.dev}(\tilde{Y}|X=\alpha), \text{st.dev}(\tilde{Y}|X=\beta)\right\}}\right\}$ ;
4 $\boldsymbol{x} \leftarrow \left[\alpha, \alpha + \frac{\beta-\alpha}{\tau-1}, \alpha + 2\,\frac{\beta-\alpha}{\tau-1}, \ldots, \beta\right]$, $\boldsymbol{w} \leftarrow \frac{\beta-\alpha}{\tau}\left[\frac{1}{2}, 1 \ldots, 1, \frac{1}{2}\right]$ ;
5 return $\sum_{i=1}^{\tau} w_i \cdot f(x_i) \cdot \tilde{f}(y_0|x_i)$;
```

Algorithm 2: Skeleton for the *uniform fill-in* algorithm.

We will end this part by a small simulation study. Consider again the model $X \rightarrow Y$, and let X follow a Normal distribution with mean zero and variance σ_X^2. Further, let $Y|\{X = x\} \sim N(x, \sigma_Y^2)$. It follows that the marginal distribution for Y is $N(0, \sigma_X^2 + \sigma_Y^2)$. In Fig. 6 (a) we show the calculated marginal distribution of Y using Algorithm 1 with $\sigma_X^2 = 10^{10}$ and $\sigma_Y^2 = 10^{-6}$. We chose $\tau = 8$ in the numerical integration scheme, and conclude that the results are far from the sought solution. The results using adaptive selection of integration points (bounded above by $\tau_{\max} = 16$) and uniform fill-in are shown in Fig. 6 (b). The marginal density plot is quite close to the true model, and it is evident that the adaptions to the algorithm have removed the numerical problems the vanilla-version of the algorithm were suffering from.

4 Conclusions

In this paper we have discussed an approximative scheme for dynamic discretisation. The method was published in [5], and although it has been well received



(a)Vanilla version (b) Uniform fill-in and adaptive level selection

Fig. 6. The plots show the marginal distribution for Y after discretisation. Part (a) gives the results of Algorithm 1, whereas Part (b) shows the results when utilising uniform fill-in and the adaptive technique for choosing the level of the numerical integration.

it has not been given a formal evaluation before now. Its distinguishing feature is that each variable is discretised based only on its marginal posterior distribution, which is in contrast to the original dynamic discretisation algorithm by [3], where the all variables in a clique are analysed together.

The simplification can potentially lead to remarkable speed-ups during inference (Kozlov and Koller's algorithm is about ten times slower than Algorithm 1 on an example network consisting of only two variables). On the other hand, we have explained why the results of the simplified approach may sometimes give inferior results. Sub-optimal discretisation is particularly evident for a variable having a large marginal variance compared to its conditional variance. We analysed this problem in detail and proposed a solution which significantly improves the quality of the results. In conclusion, our results suggest that the approximate procedure with simple extensions is feasible also for very challenging problems.

References

1. Shenoy, P.P., Shafer, G.: Axioms for probability and belief-function propagation. In: Proceedings of the Sixth Workshop on Uncertainty in Artificial Intelligence. (1990) 169–198
2. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and unsupervised discretization of continuous features. In: Proceedings of the Twelfth International Conference on Machine Learning. (1995) 194–202
3. Kozlov, A.V., Koller, D.: Nonuniform dynamic discretization in hybrid networks. In: Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence. (1997) 314–325
4. Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley & Sons, New York (1991)
5. Neil, M., Tailor, M., Marquez, D.: Inference in Bayesian networks using dynamic discretisation. Statistics and Computing **17**(3) (2007) 219–233
6. Neal, R.: Sampling from multimodal distributions using tempered transitions. Statistics and Computing **6** (1996) 353–366