Predictive Discretization during Model Selection

Harald Steck
Computer-Aided Diagnosis and Therapy
Siemens Medical Solutions, 51 Valley Stream Parkway E51
Malvern, PA 19355, USA
harald.steck@siemens.com

Tommi S. Jaakkola
MIT CSAIL,
Stata Center, Bldg. 32-Gates 498,
Cambridge, MA 02139, USA
tommi@csail.mit.edu

Abstract

We present an approach to discretizing multivariate continuous data while learning the structure of a graphical model. We derive the joint scoring function from the principle of predictive accuracy, which inherently ensures the optimal trade-off between goodness of fit and model complexity (including the number of discretization levels). Using the so-called finest grid implied by the data, our scoring function depends only on the number of data points in the various discretization levels. Not only can it be computed efficiently, but it is also invariant under monotonic transformations of the continuous space. Our experiments show that the discretization method can substantially impact the resulting graph structure.

1 Introduction

Continuous data is often discretized as part of a more advanced approach to data analysis, like, e.g., structure learning in graphical models. Discretization may be carried out merely for computational efficiency, or because background knowledge suggests that the underlying variables are indeed discrete. While it is computationally efficient to discretize the data in a preprocessing step that is independent of the subsequent analysis, e.g., [6, 11, 7], the impact of the discretization policy on the subsequent analysis typically remains unknown in this approach. For this reason, methods have been developed that optimize the discretization policy and the graphical model *jointly*, e.g., [3, 10]. However, the proposed algorithms are computationally very involved, prohibiting their application to reasonably large real-world data sets.

We derive a novel scoring function that (1) allows one to optimize the discretization policy and the structure of the graphical model jointly, and (2) can be computed efficiently. We adopt predictive accuracy as the objective, as this inherently ensures the optimal trade-off between model fit and model complexity. The two most common ways of assessing predictive accuracy are cross-validation [17], and sequential approaches like prequential validation or stochastic complexity [2, 13]. We focus on the predictive sequential approach in this paper in the interest of space, and omit our semi-predictive approach as well as the scoring function based on cross-validation (cf. [16] for details).

In the next section, we present the basic sequential approach. Section 3 introduces the *finest grid implied by the data*, which is conceptually essential for the simplicity of our result. In Section 4, we derive the scoring function for predictive discretization and discuss its properties. Finally, we show in our experiments in Section 6 that discretization can indeed have a crucial impact on the resulting graph structure.

2 Sequential Approach

In this section, we introduce the basic idea of sequential assessment of predictive accuracy, together with relevant notation. Let the n continuous variables in the domain of interest be $Y = (Y_1, ..., Y_k, ..., Y_n)$, and their instantiation y. The discretization of the continuous variable Y is determined by the *discretization policy* $\Lambda = (\Lambda_1, ..., \Lambda_n)$. Concerning each variable Y_k, let $\Lambda_k = (\lambda_{k,1}, ..., \lambda_{k,r_k-1})$ denote the discretization sequence such that $\lambda_{k,j} < \lambda_{k,j+1}$ for all $j = 1, ..., r_k - 2$, where r_k is the number of discretization levels. This determines the mapping $f_\Lambda : Y \mapsto X$, where $X = (X_1, ..., X_k, ..., X_n)$ is the corresponding discretized vector:

$$f_{\Lambda_k}(y_k) = \begin{cases} 1 & \text{if } y_k < \lambda_{k,1} \\ j & \text{if } \lambda_{k,j-1} \leq y_k < \lambda_{k,j} \quad \text{for } 1 < j < r_k \\ r_k & \text{if } \lambda_{k,r_k-1} \leq y_k \end{cases} \qquad (1)$$

For computational efficiency, we consider *deterministic* discretization throughout this paper, i.e., each continuous value y is mapped to *exactly one* discretization level, $x_k = f_{\Lambda_k}(y_k)$.

In our sequential approach, we pretend that (continuous) *i.i.d.* data D arrive in a sequential manner, and then assess predictive accuracy regarding each data point along the sequence. This is similar in spirit to prequential validation or stochastic complexity [2, 13]. We recast the joint marginal likelihood of the discretization policy Λ and the structure m of a graphical model in a sequential manner,

$$\rho(D|\Lambda, m) = \prod_{i=1}^{N} \rho(y^{(i)}|D^{(i-1)}, \Lambda, m), \qquad (2)$$

where $D^{(i-1)} = (y^{(i-1)}, y^{(i-2)}, ..., y^{(1)})$ denotes the data points seen *prior to* step i along the sequence. Any sequential ordering of the data points may be chosen for *i.i.d.* data D, lacking a natural ordering. While the value of $\rho(D|\Lambda, m)$ may depend on the chosen sequential ordering, we outline in Section 4.2 that it is independent of the ordering in good approximation for reasonably large data sets. Eq. 2 shows that high predictive accuracy is inherently tied to a large marginal likelihood $\rho(D|\Lambda, m)$. Assuming deterministic discretization, at each step i the predicted density regarding data point $y^{(i)}$ factors,

$$\rho(y^{(i)}|D^{(i-1)}, \Lambda, m) = \rho(y^{(i)}|x^{(i)}, \Lambda)\ p(x^{(i)}|D^{(i-1)}, m, \Lambda),$$

where $x^{(i)} = f_\Lambda(y^{(i)})$ according to Eq. 1. When learning the structure m of a graphical model, it is desirable that m indeed captures *all* the relevant (conditional) dependences among the variables $Y_1, ..., Y_n$. We thus make the assumption that the dependences among the continuous variables Y_k are described by the underlying *discretized* distribution $p(X|m, \Lambda, D)$; hence, any two continuous variables Y_k and $Y_{k'}$ are independent conditional on X,

$$\rho(y^{(i)}|x^{(i)}, \Lambda) = \prod_{k=1}^{n} \rho(y_k^{(i)}|x^{(i)}, \Lambda_k). \qquad (3)$$

The computational feasibility of this approach depends crucially on the efficiency of the mapping between the discretized space X and the continuous one, Y. The simplest approach is obtained by assigning the *same* density to all the points y and y' that are mapped to the same discretized state x, cf., e.g., [10]. Assuming such a *uniform* probability density is a stringent restriction on Eq. 3, as the latter requires only *independence* of the variables Y_k. When the data points y are distributed non-uniformly according to $y \sim \prod_{k=1}^{n} \rho(Y_k|x, \Lambda_k)$, the use of a uniform density needlessly degrades the predictive accuracy.

3 Finest Grid implied by the Data

The *finest grid implied by the data* provides a simple mapping that (1) retains the desired independence properties according to Eq. 3, allowing for non-uniform densities, and (2) can be computed efficiently. It provides an *implicit* estimate of the densities $\rho(y_k^{(i)}|x^{(i)}, \Lambda_k)$ in Eq. 3. While this grid is conceptually important for deriving our predictive scoring function, note that the resulting Eq. 9 turns out to be (approximately) independent of it.

Roughly speaking, this grid is obtained by discretizing each continuous variable Y_k $(k = 1, ..., n)$ such that there is *exactly one* data point in each discretization level.We denote the discretization policy associated with the finest grid by $\Omega = (\Omega_1, ..., \Omega_n)$, where the discretization sequence $\Omega_k = (\omega_{k,1}, ..., \omega_{k,N-1})$ is such that, for all $j = 1, ..., N-2$, we have $\omega_{k,j} < y_k^{(i)} < \omega_{k,j+1}$ for exactly one $y_k^{(i)}$; N is the number of data points.[1] The threshold values $\omega_{k,j}$ may be chosen to be *any* value between neighboring data points.[2] Note that the finest grid is not unique because of this freedom in the choice of the threshold values. Analogously to Eq. 1, the discretization policy Ω implies a deterministic mapping to a new vector of discrete random variables, say Z, $f_\Omega : Y \mapsto Z$. Moreover, this also determines the mapping $f_{\Omega,\Lambda} : Z \mapsto X$ between two *discrete* spaces, based on the discretization policies Λ and Ω

Let us introduce further notation for later use: let $[z_k]_{\Omega_k} = [\omega_{k,z_k-1}, \omega_{k,z_k})$ for each $z_k = 1, ..., N$ denote the intervals (in the continuous space) according to the finest grid;[3] and $[z]_\Omega = \times_{k=1}^{n} [z_k]_{\Omega_k}$ the hyperrectangles in the n-dimensional space.

4 Predictive Discretization

Predictive discretization leads to a fair score, as the density at data point $y^{(i)}$ is predicted strictly without hindsight at each step i, i.e., only data $D^{(i-1)}$ is used. Before we derive our new predictive scoring function in Section 4.2, we first outline how the finest grid changes along the sequence, as it can only be based on the data $D^{(i-1)}$ seen prior to each step i.

[1] For simplicity, we pretend that the grid is based on D (with N data points) when introducing notation here. In fact, the grid is based only on data $D^{(i-1)}$ at each step i of our sequential approach, as outlined in detail in Section 4.1.

[2] As a special case, e.g., the midpoints may be selected.

[3] We define $\omega_{k,0} = a_k$ and $\omega_{k,N} = b_k$ when Y_k takes on values in $[a_k, b_k]$; a_k, b_k finite.

4.1 Time-Evolution of Finest Grid

Our objective is to assess the predictive accuracy of the pair (Λ, m) vs. the pair (Λ', m'). We use two different finest grids, each of which pertaining to (Λ, m) and (Λ', m'), respectively. In the following, we specify how $\Omega_\Lambda^{(i-1)}$, i.e., the finest grid pertaining to (Λ, m), evolves along the sequence $(i = 1, ..., N)$; the other grid, and hence $\Omega_{\Lambda'}^{(i-1)}$, is defined analogously.

At $i = 1$, i.e., before any data is seen, let the finest grid pertaining to the pair (Λ, m) be identical to the grid implied by Λ, i.e., $\Omega_\Lambda^{(0)} = \Lambda$. Note that there is exactly one hyper-rectangle $[z]_{\Omega_\Lambda^{(0)}}$ that is mapped to each x, although there is no data point in any of the hyper-rectangles $[z]_{\Omega_\Lambda^{(0)}}$ at this point.

As we proceed along the sequence, we update $\Omega_\Lambda^{(i-1)}$ in order to obtain $\Omega_\Lambda^{(i)}$ as follows: if $y_k^{(i)}$ lies in an interval $[z_k]_{\Omega_{\Lambda,k}^{(i-1)}}$ that already contains a data point, then a new threshold value is introduced that splits $[z_k]_{\Omega_{\Lambda,k}^{(i-1)}}$ into two new intervals, $[z_k]_{\Omega_{\Lambda,k}^{(i)}}$ and $[z'_k]_{\Omega_{\Lambda,k}^{(i)}}$, each of which containing *exactly one* data point (for all $k = 1, ..., n$). Note that there is the freedom of choosing any particular threshold value between the neighboring data points, so that we can select that value as follows: if we can choose a threshold value that coincides with one of the threshold values of the other discretization policy Λ', we do so; otherwise, we choose any, but the *same* threshold value for both $\Omega_\Lambda^{(i)}$ and $\Omega_{\Lambda'}^{(i)}$. Due to this choice of threshold values, there exists a (rather small) $i_0 \leq N$ such that $\Omega_\Lambda^{(i)} = \Omega_{\Lambda'}^{(i)}$ for all $i \geq i_0$, while $\Omega_\Lambda^{(i)} \neq \Omega_{\Lambda'}^{(i)}$ for $i = 1, ..., i_0 - 1$. Obviously, the value of i_0 depends on the particular sequential ordering of the data points. Since *i.i.d.* data lack an inherent sequential ordering, we may choose a *particular* ordering of the data points. This is similar in spirit to stochastic complexity [13], where also a *particular* sequential ordering is used. Our aim is to choose such a sequential ordering that minimizes i_0 when we compare the pairs (Λ, m) and (Λ', m') to each other: we require that, during a *short* initial phase, at least one data point is assigned to *each* discretization level pertaining to the *joint* discretization policy $\Lambda^\cup$ of Λ and Λ'.[4,5]Hence, we have the bound $i_0 \leq \max_k(|X_k|_\Lambda) + \max_k(|X_k|_{\Lambda'})$, where $|\cdot|_{\Lambda/\Lambda'}$ denotes the number of discretization levels of X_k due to Λ and Λ', respectively. With the further assumption that the number of discretization levels is bounded from above, we have $i_0 \ll N$ given a reasonably large data set D. For $i \geq i_0$, we permit an arbitrary sequential ordering, as we have $\Omega_\Lambda^{(i)} = \Omega_{\Lambda'}^{(i)}$.

4.2 Predictive Scoring Function

Concerning the pair (Λ, m), and analogously for (Λ', m'), we can now obtain an efficient mapping between Y and X, namely via Z at each step i: each term in Eq. 3 decomposes like

$$\begin{aligned}&\rho(y_k^{(i)}|x^{(i)}, \Lambda_k, \Omega_{\Lambda,k}^{(i-1)})\\ &= \rho(y_k^{(i)}|z_k^{(i)}, \Omega_{\Lambda,k}^{(i-1)})\ p(z_k^{(i)}|x^{(i)}, \Lambda_k, \Omega_{\Lambda,k}^{(i-1)}). \quad (4)\end{aligned}$$

Regarding the mapping between Y and Z, we allow for any strictly positive density $\rho(Y_k|Z_k, \Omega_{\Lambda,k}^{(i-1)})$. In order to *efficiently* map the probability mass predicted for $x^{(i)}$ to the finest grid (Z) we make one more simplification, namely that the probability mass predicted for $x^{(i)}$ is divided *evenly* among all the hyper-rectangles $[z]_{\Omega_\Lambda^{(i-1)}}$ that are mapped to $x^{(i)}$, irrespectively of their possibly different volumes. This simplification can be motivated as follows: the definition of the finest grid implied by the data entails immediately that the volumes of the hyper-rectangles $[z]_{\Omega_\Lambda^{(i-1)}}$ tend to be larger in those regions of the continuous space where the data points $y^{(1)}, ..., y^{(i-1)}$ are sparser; hence, this mapping automatically tends to predict a lower probability for regions with a lower density of data points, which is desirable. With this assumption, we immediately obtain

$$p(z_k^{(i)}|x^{(i)}, \Lambda_k, \Omega_{\Lambda,k}^{(i-1)}) = \frac{1}{N_+^{(i-1)}(x_k^{(i)})}, \quad (5)$$

where $x_k^{(i)} = f_{\Omega_k,\Lambda_k}(z_k^{(i)})$; $N_+^{(i-1)}(x_k^{(i)}) = \max\{1, N^{(i-1)}(x_k^{(i)})\}$ arises from the fact that, at small $i \leq i_0$, there is at *least one* hyper-rectangle mapped to each $x^{(i)}$, even if it does not contain a data point (yet);[6] $N^{(i-1)}(\cdot)$ denotes the counts based on the discretized data $D_\Lambda^{(i-1)}$ seen *before* step i. This is because the finest grid, and hence $\Omega_\Lambda^{(i-1)}$, is based on $D^{(i-1)}$.

Our predictive scoring function for the pair (Λ, m) follows now immediately by substituting the above equations into Eq. 2:

$$\rho(D|\Lambda, m) = \frac{p(D_\Lambda|m)}{G(D, \Lambda)} \cdot \left(\prod_{i=1}^{N} \prod_{k=1}^{n} \rho(y_k^{(i)}|z_k^{(i)}, \Omega_{\Lambda,k}^{(i-1)}) \right), \quad (6)$$

where $z_k^{(i)} = f_{\Omega_k}(y_k^{(i)})$. Some comments on each of the three terms are in order. $p(D_\Lambda|m)$ is the marginal

[4] For $k = 1, ..., n$: $\Lambda_k^\cup$ comprises the threshold values of *both* Λ_k and Λ'_k.

[5] This entails a (slight) restriction on Λ and Λ', as they have to be such that there is at least one data point in each bin pertaining to their joint discretization policy $\Lambda^\cup$. However, this restriction can be resolved [16].

[6] Note that this is different from using a prior (e.g., unlike in [1, 8]).

likelihood of the graph m in light of the data D_Λ discretized according to Λ. In a Bayesian approach, it can be calculated easily for various graphical models, e.g., see [1, 8] concerning discrete Bayesian networks. $G(D, \Lambda)$ originates from Eq. 5, i.e., from the mapping between X (discretized by Λ) and the finest grid (Z) and reads

$$G(D, \Lambda) = \prod_{k=1}^{n} \prod_{x_k} \Gamma(N(x_k)). \tag{7}$$

The Gamma function, $\Gamma(N(x_k)) = [N(x_k) - 1]!$, is well-defined because $N(x_k) = N_+(x_k) \geq 1$ due to our assumption in footnote 5. Note that Eq. 7 is independent of the finest grid. The third term in Eq. 6 is due to the mapping between the finest grid (Z) and the continuous space (Y). It can be decomposed like

$$\left(\prod_{i=i_0+1}^{N} \rho(y^{(i)}|z^{(i)}, \Omega_\Lambda^{(i-1)}) \right) \cdot \left(\prod_{i=1}^{i_0} \rho(y^{(i)}|z^{(i)}, \Omega_\Lambda^{(i-1)}) \right), \tag{8}$$

which depends on the exact sequential ordering. However, the first term ($i > i_0$) is *identical* concerning both Λ and Λ', and is hence irrelevant when comparing those two discretization policies to each other. Due to the second term in Eq. 8, our predictive scoring function hence depends only on the sequential ordering during a (short) initial phase ($i \leq i_0$). Because of $i_0 \ll N$ (cf. Section 4.1), the second term in Eq. 8 becomes negligible compared to the terms that grow with N for large N. Given a reasonably large data set ($\max_k(|X_k|_\Lambda) + \max_k(|X_k|_{\Lambda'}) \ll N$), we can thus obtain a good approximation by ignoring the second term in Eq. 8 as well.

Ignoring irrelevant terms, we obtain the (approximate) predictive scoring function for the pair (Λ, m), and analogously for (Λ', m'):

$$\mathcal{L}_\text{P}(\Lambda, m) = \log p(D_\Lambda|m) - \log G(D, \Lambda). \tag{9}$$

This scoring function has several interesting properties (cf. [16] for more details). First, Eq. 9 is an *absolute* scoring function of (Λ, m), i.e., it is independent of (Λ', m'). This allows us to compare *several* discretization policies directly to each other, irrespective of the underlying fact that each pair is possibly compared with respect to a different sequential ordering. Second, the difference between $\log p(D_\Lambda|m)$ and $\log G(D, \Lambda)$ determines the trade-off dictating the optimal number of discretization levels, threshold values and graph structure. As both terms increase with a diminishing number of discretization levels, $\log G(D, \Lambda)$ can be viewed as a penalty for small numbers of discretization levels. Third, as expected for *i.i.d.* data, the resulting scoring function $\mathcal{L}_\text{P}(\Lambda, m)$ is independent of the particular ordering chosen in our sequential approach. Fourth, $\mathcal{L}_\text{P}(\Lambda, m)$ depends on the number of data points in the different discretization levels only. This has several interesting implications. First, all discretization policies that lead to the same number of data points in each discretization level, but possibly differ in the particular threshold values, are assigned the same score (and are hence *equivalent* w.r.t. our scoring function). Second, this approach includes as a special case quantile discretization, namely when all the variables are independent of each other ($m = m_\text{empty}$). The number of states is then chosen to optimize predictive accuracy (one state being optimal unless constraints are imposed). Third, and most important from a practical point of view, it renders efficient computations possible: as the search space of Λ's is huge, it is particularly important that the scoring function can be evaluated efficiently for any Λ during the search process. Fourth, $\mathcal{L}_\text{P}(\Lambda, m)$ is independent of the particular choice of the finest grid. Fifth, $\mathcal{L}_\text{P}(\Lambda, m)$ is invariant under monotonic transformations of the continuous variables. Obviously, this can lead to considerable loss of information, particularly when the (Euclidean) *distances* among the various data points in the continuous space govern the discretization. On the other hand, the results of our scoring function are not degraded if the data is given w.r.t. an inappropriate metric. In fact, the optimal discretization w.r.t. our scoring function is based on *statistical dependence* of the variables, rather than on the distances w.r.t. the *metric* (cf. [16] for further details).

5 Optimizing our Scoring Function

This section provides only the main steps of our heuristic aimed to find the maximum of our scoring function (Eq. 9), as the focus of this paper is on the scoring function itself. Instead of using a search strategy in the *joint* space of graphs and discretization policies — the theoretically best, but computationally most involved approach — we optimize the graph m and the discretization policy Λ alternately in a greedy way for simplicity: given the discretized data D_Λ, we use local search to optimize the graph m, like in [8]; given m, we optimize Λ iteratively by improving the discretization policy regarding a *single* variable given its Markov blanket at a time. The latter optimization is carried out in a hierarchical way over the number of discretization levels and over the threshold values of each variable. Local maxima are a major issue when optimizing the predictive scoring function due to the (strong) interdependence between m and Λ. As a simple heuristic, we alternately optimize Λ and m only slightly at each step.

6 Experiments

We first present several experiments on 2-dimensional toy data, as this allows us to visualize the data points for validation of the learned discretization policy and to explore the properties of our predictive scoring function in a well-defined setting. Finally, we apply our approach to high-dimensional gene-expression data.

Our first experiment shows that our predictive scoring function can indeed identify the correct number of discretization levels. We generated data sets where the number of clusters varied between 2 and 10, each of which contained 100 points. The data with all 10 clusters present is sketched in Figure 1. Obviously, the optimal discretization policy is such that every cluster is assigned to a separate state. This is indeed favored by our predictive score, as shown in Figure 2: the correct number of discretization levels obtained the highest predictive score. Moreover, Figure 2 shows that, when the score is optimized, it decreased quickly to the left of the correct number of discretization levels, while it dropped quite slowly to its right. The reason for the slow decrease to the right is that the 'excess' levels beyond the optimum number are almost empty (our approach forbids that they are completely empty) due to the optimization of the threshold values, i.e., the resulting discretization is very similar to the optimal one. Even though the decrease to the right of the optimum is less pronounced than to the left, the decrease still appears significant: for instance, given 5 clusters in the data, the predictive scores of 4, 5, 6, 7, 8, 9 and 10 levels are 649.0, 783.9, 776.5, 774.0, 764.8, 758.8 and 752.2, respectively. Note that a difference in the log-score of 3 is considered 'strong', and 5 'very strong' evidence in the statistics literature, e.g., [9]. Concerning the other data sets with 2,...,10 clusters, the log-score of the correct number of clusters was also supported by 'very strong' evidence over both the smaller and larger numbers of discretization levels. The reason for the steep decrease to the left of the optimum is that the optimum choice of threshold values cannot compensate much for having too few discretization levels available. When the threshold values do not get optimized, cf. dashed lines in Figure 2), the predictive score drops considerably both to the right and to the left of the optimum, as expected. Moreover, when the fraction $1/r$ of points gets assigned to each of the r discretization levels ($r = 2, ..., 10$), the number of states that are multiples or divisors of the optimal number tend to be local optima of the predictive score, as expected. Note that this simple heuristic finds the optimum w.r.t. our predictive scoring function in these toy data sets simply because each cluster contains the same number of points.

The second experiment illustrates that the optimum discretization w.r.t. our predictive scoring function is based on statistical dependence between the variables rather than on clusters in the metric space. Consider the top two panels in Fig. 3: when the variables are *independent*, our approach may not find the discretization suggested by the clusters; instead, our approach assigns the same number of data points to each discretization level (with the minimum number of discretization levels being optimal). Note that discretization of independent variables is, however, quite irrelevant when learning graphical models: the optimal discretization of each variable Y_k depends on the variables in its Markov blanket, and Y_k is (typically strongly) dependent on those variables. When the variables are *dependent* in Fig. 3 (top right), our scoring function favors the "correct" discretization (solid lines), as this entails best predictive accuracy (even when disregarding the metric). However, dependence of the variables itself does not necessarily ensure that our scoring function favors the "correct" discretization, as illustrated in the bottom two panels in Fig. 3 (as a constraint, we require two discretization levels): given low noise levels, our scoring function assigns the same number of data points to each discretization level; however, a sufficiently *high* noise level in the data can actually be beneficial, permitting our approach to find the "correct" discretization, cf. Fig. 3 (bottom right).

While the clusters were well-separated on the top right panel in Figure 3, Figure 4 illustrates the situation where the clusters overlap with respect to variable Y_0. As a result, the overlap region spurs the creation of additional discretization levels. This makes sense, as separating the overlap region from the well-separated regions increases predictive accuracy in the discretized domain. Learning from re-sampled data, we noticed that the optimum number of discretization levels of Y_0 varied between 3 and 5 states for this data set, i.e., the additional states created in the overlap region varied between 1 and 3.

The marginal likelihood $p(D_\Lambda|m)$, which is part of our scoring function, contains a free parameter, namely the so-called scale-parameter α regarding the Dirichlet prior over the model parameters, e.g., cf. [8]. As outlined in [14], its value has a decisive impact on the

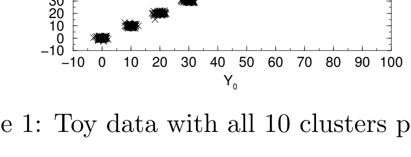

Figure 1: Toy data with all 10 clusters present.

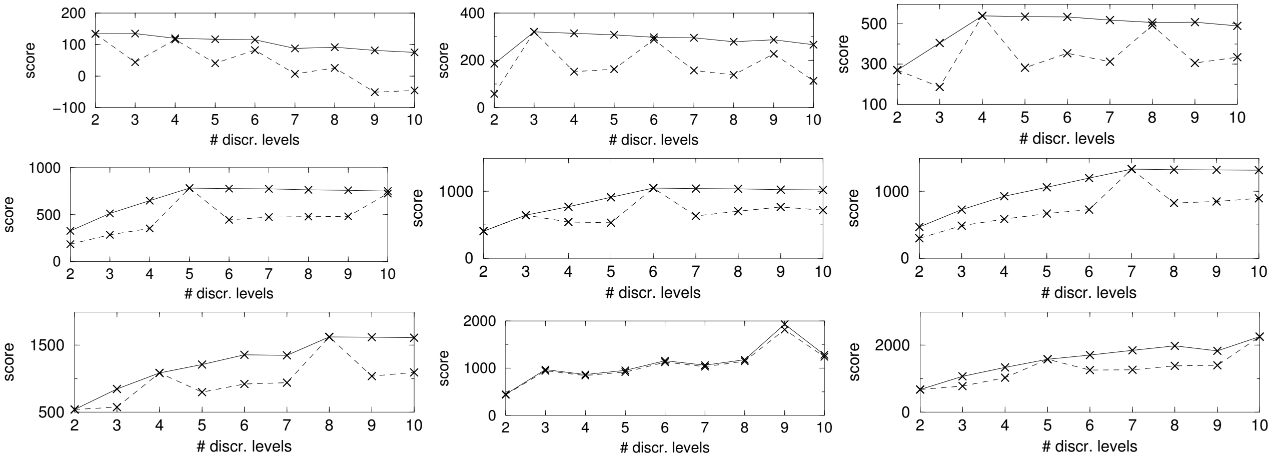

Figure 2: Each panel is based on a different data set, with the number of clusters increasing from 2 to 10 from the top left panel to the bottom right one (line by line). Each panel displays our predictive score as a function of the number of discretization levels: the maximum score after optimizing the threshold values (solid line); and the score when simply assigning the fraction $1/r$ of points to each of the $r = 2, ..., 10$ discretization levels (dashed line), i.e., without optimization of the threshold values.

resulting number of edges in the network, and must hence be chosen with care. When considering α as an additional parameter to be learned from the data, we found for the optimal values $\alpha = 1, ..., 10$ in our toy data sets. It appeared that a growing value of α tended to result in an increased number of discretization levels. However, this increase did not appear monotonic (possibly, the search heuristic got trapped in local optima of the predictive scoring function), cf. Fig. 4.

The next experiment shows that the optimal number of discretization levels in the 'overlap region' increases as the number of data points grows. In the data set depicted in the inset of Figure 5, there is only a single cluster, sampled from a joint normal distribution with correlation $corr(Y_0, Y_1) = 1/\sqrt{2}$. As this distribution does not imply a 'natural' number of discretization levels, our predictive scoring function yields the optimal 'effective' number of discretization levels, with the goal to maximize the degree of dependence between discretized variables, accounting for regularization. Fig. 5 shows that our predictive scoring function favored an increasing number of discretization levels as the sample size increased, as expected from a regularization point of view. Moreover, the learned graph structure implied independence of Y_0 and Y_1 when given very small samples (fewer than 30 data points in our experiment), while Y_0 and Y_1 are found to be dependent for all larger sample sizes. Our scoring function thus favored less complex models (i.e., sparser graphs and fewer discretization levels) when given smaller data sets. This is desirable in order to avoid overfitting when learning from small samples, leading to better predictive accuracy.

Our final experiment illustrates that the discretization policy can have a crucial impact on the learned Bayesian network structure. We re-analyzed gene expression data concerning the pheromone response pathway in yeast [7], comprising 320 measurements for 32 continuous variables (genes) as well as the mating type (binary variable). In computational biology, Bayesian networks have been used to model regulatory networks, and their structures were learned from gene-expression data discretized in a pre-processing step, e.g., [6, 12, 7]. As to account for model uncertainty due to the small data set, we used a non-parametric re-sampling method instead of Markov Chain Monte Carlo methods, as the former is independent of any model assumptions. While the bootstrap has been used in [5, 4, 6, 12], we prefer the jackknife when learning the graph structure, i.e., conditional independences. The reason is that the bootstrap procedure can easily induce spurious dependencies when given a small data set D; as a consequence, the resulting network structure can be considerably biased towards denser graphs [15]. The jackknife avoids this problem. We obtained very similar results using three different variants of the jackknife: delete-1, delete-30, and delete-64. Assessing predictive accuracy by means of 5-fold cross validation, we determined the optimal scale-parameter of the Dirichlet prior to be $\alpha \approx 25$ in our scoring function.

Using the optimal value of α, Figure 6 summarizes the learned network structures for different discretization policies: when the number of discretization levels is

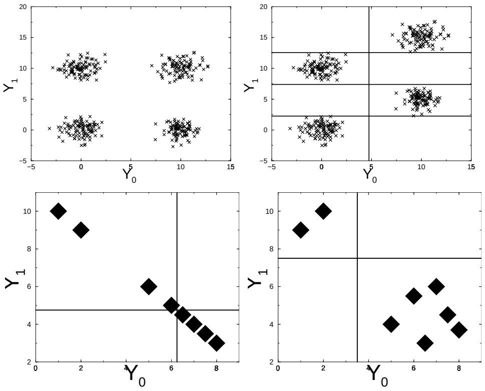

Figure 3: Top two panels: each cluster comprises 100 points sampled from a Gaussian distribution; Y_0 and Y_1 are independent on the left, and dependent on the right. Bottom two panels: when Y_0 and Y_1 are dependent, *noise* may help in finding the 'correct' discretization.

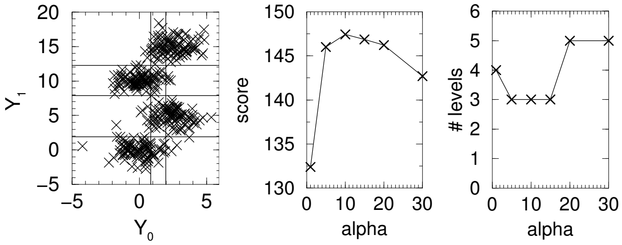

Figure 4: 4 overlapping clusters with 100 points each: the optimal discretization (left); the other two panels show the dependence of the maximum score and of the number of discretization levels of Y_0 on the scale-parameter α of the Dirichlet prior, cf. text for details.

not optimized, but set to a fixed number r (which we chose to be identical for all the variables), the number of edges in the network increases with a decreasing value of r. The reason is that the (standard) posterior probability of discrete Bayesian network structures implicitly contains a penalty term for model complexity that grows with the number of independent variables in the model (cf. Bayesian Information Criterion): assuming that all discretized variables X_k have the same number of states r for simplicity, this number is given by $\sum_k (r-1)\cdot r^{\#\ \text{parents}_k}$; as it increases exponentially with the number of parents, a large number of discretization levels thus entails the model complexity to grow quickly with the number of edges, forcing the network structure to be extremely sparse. Moreover, when the threshold values are optimized w.r.t. our predictive scoring function, then the learned number of edges is consistently larger—thus recovering more relevant dependences—than the one obtained by assigning a fraction $1/r$ of points to each discretization level (i.e., without optimizing the threshold values). Moreover, when both the number of discretization levels and the threshold values are optimized, our predictive scoring function yields 65.7 ± 8 edges; most of the variables had about 4 discretization levels (on average over the jackknife samples), except for the genes MCM1, MFALPHA1, KSS1, STE5, STE11, STE20, STE50, SWI1, TUP1 with about 3 states, and the genes BAR1, MFA1, MFA2, STE2, STE6 with ca. 5 states. Apart from that, the increase in the standard deviation indicates that our (simple) heuristic search strategy may get trapped in local optima in the different jackknife samples.

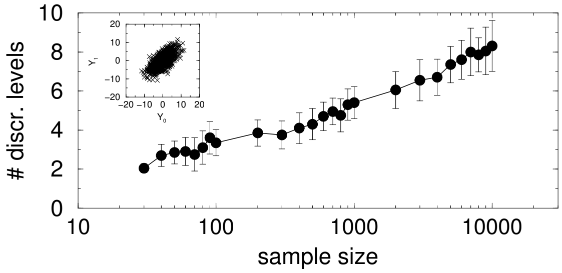

Figure 5: The number of discretization levels (mean and standard deviation, averaged over 10 samples of each size) depends on the sample size (cf. text for details).

Figure 7 shows the composite graph we learned from the gene expression data, employing our predictive scoring function, cf. Eq. 9.[7] The graph is compiled by averaging over several Bayesian network structures in order to account for model uncertainty: the solid ones are present with probability $> 50\%$, and the dashed ones with probability $> 34\%$. The orientation of an edge is indicated only if one direction is at least twice as likely as the contrary one. The crucial impact of the used discretization policy Λ and scale-parameter α on the resulting network structure becomes apparent when our network structure are compared to the one reported in [7]: their network structure resembles a naive Bayesian network, where the mating type is the root variable. Obviously, their network structure is notably different from ours in Figure 7, and hence has very different (biological) implications. Unlike in [7], we have optimized the discretization policy Λ and the network structure m jointly, as well as the scale-parameter α. As the value of the scale-parameter α mainly affects the *number* of edges present in the learned graph [14], this suggests that the major differences in the obtained network structures are actually due to the discretization policy.

[7] We imposed no constraints on the network structure in Figure 7.

#	thresholds optimized	
levels	no	yes
3	68.5 ± 2	76.8 ± 3
4	50.7 ± 2	59.2 ± 3
5	39.8 ± 2	44.0 ± 3

Figure 6: Number of edges(±std) in learned network structure for different discretization policies (average of 100 delete-30 jackknife samples). See text for details.

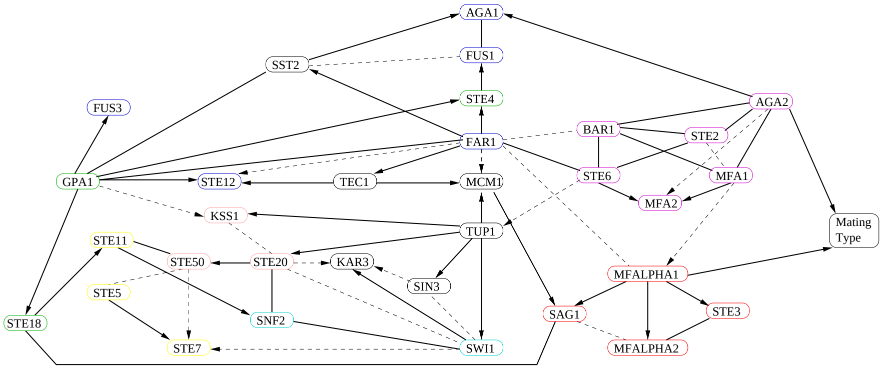

Figure 7: This graph is compiled from 320 delete-30 jackknife samples (cf. [7] for the color-coding).

7 Conclusions

We have derived a simple, yet principled and computationally efficient approach for determining the resolution at which to represent continuous observations. Our new scoring function relies on predictive accuracy in the prequential sense and employs the so-called finest grid implied by the data as the basis for finding the appropriate levels. Our experiments show its crucial impact on both the learned discretization policy as well as on the resulting graph structure.

Acknowledgements

We would like to thank Alexander Hartemink for making the pheromone data available to us. Harald Steck is grateful to Bharat Rao for encouragement and support, and to the German Research Foundation (DFG), grant STE 1045/1-2. Tommi Jaakkola acknowledges support from NSF ITR grant 0428715.

References

[1] G. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–47, 1992.

[2] A. P. Dawid. Statistical theory. The prequential approach. *J.R.Stat.Soc.Ser.A*, 147:277–305,1984.

[3] N. Friedman and M. Goldszmidt. Discretizing continuous attributes while learning Bayesian networks. *ICML*, pp. 157–65, 1996.

[4] N. Friedman, M. Goldszmidt, and A. Wyner. Data analysis with Bayesian networks: A bootstrap approach. *UAI*, pp. 196–205, 1999.

[5] N. Friedman, M. Goldszmidt, and A. Wyner. On the application of the bootstrap for computing confidence measures on features of induced Bayesian networks. *AISTATS*, pp. 197–202, 1999.

[6] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. *J. Comp. Biol.*, 7:601–20, 2000.

[7] A. J. Hartemink, D. K. Gifford, T. S. Jaakkola, and R. A. Young. Combining location and expression data for principled discovery of genetic regulatory networks. *PSB*, 2002.

[8] D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.

[9] R. E. Kass and A. E. Raftery. Bayes factors. *JASA*, 90:773–96, 1995.

[10] S. Monti and G. F. Cooper. A multivariate discretization method for learning Bayesian networks from mixed data. *UAI*, pp.404–13,1998.

[11] S. Monti and G. F. Cooper. A latent variable model for multivariate discretization. *AISTATS*, pp. 249–54, 1999.

[12] D. Pe'er, A. Regev, G. Elidan, and N. Friedman. Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 1:1–9, 2001.

[13] J. Rissanen. Stochastic complexity and modeling. *The Annals of Statistics*, 14:1080–100, 1986.

[14] H. Steck and T. S. Jaakkola. On the Dirichlet prior and Bayesian regularization. *NIPS*, 2002.

[15] H. Steck and T. S. Jaakkola. Bias-corrected bootstrap and model uncertainty. *NIPS*, 2003.

[16] H. Steck and T. S. Jaakkola. (Semi-)predictive discretization during model selection. *AI Memo 2003-002, MIT*, 2003.

[17] M. Stone. Cross-validatory choice and assessment of statistical predictions. *J. R. Stat. Soc. Ser. B*, 36:111–47, 1974.