massachusetts institute of technology — artificial intelligence laboratory

(Semi-)Predictive Discretization During Model Selection

Harald Steck and Tommi S. Jaakkola

AI Memo 2003-002 Feburary 2003

Abstract

In this paper, 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 independent of the metric used in the continuous space. Our experiments with gene expression data show that discretization plays a crucial role regarding the resulting network structure.

This work was supported by the German Research Foundation (DFG) under grant STE 1045/1-2, Nippon Telegraph and Telephone Corporation, NSF ITR grant IIS-0085836, and the Sloan Foundation in the form of the Sloan Research Fellowship.

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, 10, 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, 9]. 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) renders efficient computations possible. We adopt the objective of optimizing predictive accuracy, 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 [14], and sequential approaches like prequential validation or stochastic complexity [2, 12]. We derive scoring functions for both cases in this paper.

In the next section, we present the basic idea of the sequential approach. Section 3 introduces the *finest grid implied by the data*, which is pivotal for the computational efficiency of our approach. In Sections 4 and 5, we derive scoring functions for (semi-)predictive discretization and discuss their properties. Our approach based on cross validation is outlined in the Appendix. 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:

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

For computational efficiency, we assume *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, 12]. We recast the joint marginal likelihood of the discretization policy Λ and the structure m of a graphical model in a sequential manner,

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

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. 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,

$$\rho(y^{(i)}|D^{(i-1)}, \Lambda, m) = \rho(y^{(i)}|x^{(i)}, \Lambda)\ p(x^{(i)}|D^{(i-1)}, m, \Lambda), \tag{3}$$

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 $Y_1, ..., Y_n$. Assuming that the dependences among the continuous variables Y_k are described by the underlying *discretized* distribution $p(X|m, \Lambda, D)$, then any two continuous variables Y_k and $Y_{k'}$ are independent conditional on X,

$$\rho(y^{(i)}|x^{(i)}, \Lambda) = \prod_{k=1}^{n} \rho(y_k^{(i)}|x^{(i)}, \Lambda_k). \tag{4}$$

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., [9]. Obviously, assuming a *uniform* probability density is a stringent restriction on Eq. 4, 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 uniform density needlessly degrades the predictive accuracy.

Assuming a uniform density may also give rise to "empty states", i.e., discretization levels that do not contain a data point. For example, consider a one-dimensional continuous variable defined on the interval $[0, 3]$, and data points that lie uniformly within the intervals $[0, 1]$ and $[2, 3]$. In order to optimize predictive accuracy using the uniform assignment, it is obviously best to use *three* discretization levels: two of the discrete states refer to the intervals $[0, 1]$ and $[2, 3]$, respectively, while the third one corresponds to the interval $]1, 2[$. The latter state is "empty", i.e., contains no data point, but is beneficial in order to predict the vanishing density between the two clusters well. Clearly, a more desirable discretization in this example would yield a *binary* variable and a threshold value somewhere between the two clusters.

3 Finest Grid implied by the Data

In this section, we present a simple mapping that (1) retains the desired independence properties according to Eq. 4, allowing for non-uniform densities, and (2) can be computed efficiently. We achieve this by *implicitly* estimating the densities $\rho(y_k^{(i)}|x^{(i)}, \Lambda_k)$ in Eq. 4, using the *finest grid implied by the data*.

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.[1] This grid depends, of course, on the data. For the moment, let us assume that the finest grid is based on the *entire* data set D (with N data points).

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 $y_k^{(i)}$. 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, $f_\Omega : Y \mapsto Z$.

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] Moreover, let $|[z_k]|_{\Omega_k} = |\omega_{k,z_k-1} - \omega_{k,z_k}|$ denote the length of the interval. Regarding the n-dimensional vector Y, let us analogously denote the hypercubes[4] by $[z]_\Omega = \times_{k=1}^n [z_k]_{\Omega_k}$, and their volumes by $|[z]|_\Omega = \prod_{k=1}^n |[z_k]|_{\Omega_k}$.

The discretization policies Λ and Ω also define the mapping $f_{\Omega,\Lambda} : Z \mapsto X$. For computational efficiency, we assume that each interval $[z_k]_{\Omega_k}$ is completely mapped to exactly one discretization level x_k, i.e., all the threshold values of Λ coincide with some of Ω.[5] Note that, if Ω is chosen before Λ, this imposes a slight restriction on the *threshold values* permissible for Λ;[6] the number of data points in each of the discretization levels is, however, not affected.

Based on the finest grid implied by the data, we can now obtain an efficient mapping between Y and X, namely via Z. The probability (density) then maps like

$$\rho(y_k^{(i)}|x^{(i)}, \Lambda_k, \Omega_k) = \rho(y_k^{(i)}|z_k^{(i)}, \Omega_k)\ p(z_k^{(i)}|x^{(i)}, \Lambda_k, \Omega_k). \tag{5}$$

When Eq. 5 is substituted into the previous equations, each density $\rho(\cdot)$ becomes conditional on Ω, i.e., on the finest grid implied by the data.

[1] It is also possible to use a more general d-grid that permits $d \in \mathbb{N}$ data points in each discretization level. Without any conceptual difficulties, one obtains exactly the same scoring function as in Eq. 10 (up to an irrelevant constant). This approach thus also allows for discrete variables Y_k.

[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 the finite interval $[a_k, b_k]$.

[4] We do not assume that all sides have the same length.

[5] More formally, for all $k = 1, ..., n$, and all $s = 1, ..., r_k - 1$, we have $\lambda_{k,s} = \omega_{k,j}$ for some $j = 1, ..., N-1$.

[6] However, note that our final scoring function is independent of the threshold values, cf. Eq. 10.

Regarding the mapping between Y and Z, we allow for any strictly positive density $\rho(Y_k|Z_k,\Omega_k)$. 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 hypercubes $[z]_\Omega$ 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 hypercubes $[z]_\Omega$ tend to be larger in those regions of the continuous space where the data points $y^{(i)}$ 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

$$p(z_k^{(i)}|x^{(i)},\Lambda_k,\Omega_k) = \frac{1}{N(x_k^{(i)})}, \tag{6}$$

where $x_k^{(i)} = f_{\Omega_k,\Lambda_k}(z_k^{(i)})$; $N(x_k^{(i)})$ is the number of data points in the discretization level $x_k^{(i)}$. Note that $N(x_k^{(i)})$ is identical with the number of intervals $[z_k]_{\Omega_k}$ that are mapped to the same $x_k^{(i)}$ (given data D, there is exactly one data point in each interval $[z_k]_{\Omega_k}$ according to the definition of the finest grid).

4 Semi-Predictive Discretization

In *semi-predictive discretization*, the mapping from X to Y is based on the finest grid implied by the *entire* data set D at each step i, as introduced in the previous section. This mapping hence involves not only data $D^{(i-1)}$ seen prior to each step i, but also (future) data points $y^{(i')}$ where $i' \geq i$. Allowing for this hindsight will lead to a (slightly) unfair assessment of predictive accuracy, so that the (quantitative) balance between model fit and model complexity (here essentially the number of discretization levels) will be slightly off in the resulting scoring function. However, note that no hindsight is used for the prediction of the probability in the discretized space X. *Predictive discretization*, which is a fair assessment using no hindsight when predicting the density $\rho(y^{(i)}|D^{(i-1)},\Lambda,m)$ at each step i, is outlined in the next section.

Our scoring function for semi-predictive discretization, $\mathcal{L}_{\mathrm{SP}}(\Lambda,m)$, follows immediately from the results in the previous section. Having chosen a finest grid based on the entire data D, we can score all the discretization policies Λ that comply with that finest grid (cf. footnote 5). Combining the above Eqs. 2-6, we obtain

$$\begin{aligned}\rho(D|\Lambda,m,\Omega) &= p(D_\Lambda|m)\cdot\left(\prod_{i=1}^{N}\prod_{k=1}^{n}\frac{1}{N(x_k^{(i)})}\right)\\ &\quad\cdot\left(\prod_{i=1}^{N}\prod_{k=1}^{n}\rho(y_k^{(i)}|z_k^{(i)},\Omega_k)\right),\end{aligned} \tag{7}$$

where $z_k^{(i)} = f_{\Omega_k}(y_k^{(i)})$. Some comments on each of the three terms are in order. The first term,

$$p(D_\Lambda|m) = \prod_{i=1}^{N} p(x^{(i)}|D^{(i-1)}, \Lambda, m), \tag{8}$$

is the marginal 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.

The second and third terms in Eq. 7 are due to the mapping from X to Z and from Z to Y, respectively. Obviously, the second term can be rewritten in terms of the maximum likelihood of the empty graph m_{empty}, as the latter represents independence among the variables,

$$\left(\prod_{i=1}^{N} \prod_{k=1}^{n} \frac{1}{N(x_k^{(i)})} \right)^{-1} = p(D_\Lambda|\hat{\theta}, m_{\text{empty}}) \cdot N^{Nn}. \tag{9}$$

The factor N^{Nn} is due to the normalization of the maximum likelihood estimates of the model parameters, $\hat{\theta}_{x_k} = \hat{p}(x_k) = N(x_k)/N$. The factor N^{Nn} is irrelevant for determining the optimal discretization policy, as it is independent of both Λ and m.

The third term in Eq. 7 is the only one that depends on the metric in the continuous space as well as on the particular threshold values chosen in the finest grid. However, this term is *independent* of both Λ and m, and is hence irrelevant when comparing different discretization policies. Ignoring the irrelevant terms (summarized as c), we hence obtain the following semi-predictive (log) scoring function,

$$\begin{aligned} \mathcal{L}_{\text{SP}}(\Lambda, m) &= \log \rho(D|\Lambda, m, \Omega) - c \\ &= \log p(D_\Lambda|m) - \log p(D_\Lambda|\hat{\theta}, m_{\text{empty}}) \\ &= \log p(D_\Lambda|m) + N \sum_{k=1}^{n} H(\hat{p}(X_k)). \end{aligned} \tag{10}$$

In the last line, the maximum likelihood of the empty graph is recast in terms of the entropy H of the empirical distributions $\hat{p}(X_k)$ regarding each variable X_k.

This scoring function has several interesting properties: First, the difference in the log likelihoods between the regularized graph m and the empty graph with unregularized parameter estimates $\hat{\theta}$ determines the trade-off dictating the optimal number of discretization levels, threshold values and graph structure. As both likelihoods increase with a diminishing number of discretization levels, the second term can be viewed as a penalty for small numbers of discretization levels. Second, as expected for *i.i.d.* data, the resulting scoring function $\mathcal{L}_{\text{SP}}(\Lambda, m)$ is independent of the particular ordering chosen in our sequential

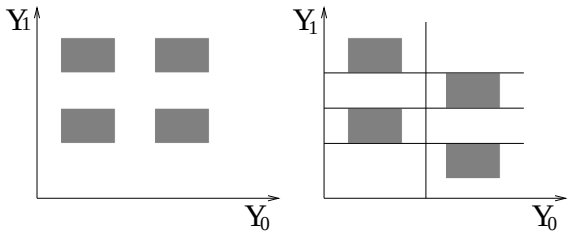

Figure 1: Assume the data points are uniformly distributed in each of the shaded rectangles.

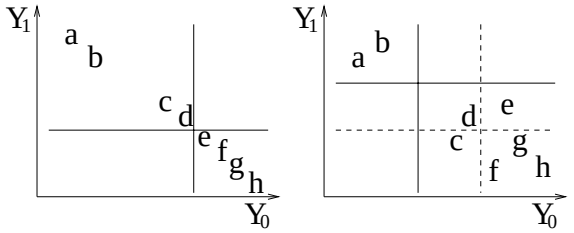

Figure 2: Based on the eight data points $(a, ..., h)$, Y_0 and Y_1 are dependent in both graphs.

approach. Third, $\mathcal{L}_{\rm SP}(\Lambda, m)$ depends on the number of data points in the different discretization levels only. This has several interesting implications. First, and most important from a practical point of view, it renders efficient computations possible. Second, and more interesting from a conceptual perspective, $\mathcal{L}_{\rm SP}(\Lambda, m)$ is independent of the particular choice of the finest grid. Third, this approach includes as a special case quantile discretization, namely when all the variables are independent of each other ($m = m_{\rm empty}$). The number of states is then chosen to optimize predictive accuracy (one state being optimal unless constraints are imposed).

Fourth, $\mathcal{L}_{\rm SP}(\Lambda, m)$ is independent of the metric in the continuous space. It is thus 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 (cf. left graphs in Figs. 1 and 2). 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 *metric*. This is illustrated in Fig. 1: 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 one discretization level

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. 1 (right graph), our scoring function favours 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 favours the "correct" discretization, as illustrated in Fig. 2 (as a constraint, we require two discretization levels): given low noise levels (left graph), our scoring function assigns the same number of data points to each discretization level. The right graph in Fig. 2 illustrates, however, that a sufficiently *high* noise level in the data can actually be beneficial, permitting our approach to find the "correct" discretization (solid lines); this is because a different discretization (dashed lines) degrades predictive accuracy, as the points c and e are assigned to "wrong" bins.

Finally, in cross validation the finest grid implied by the training data can similarly be employed when measuring predictive accuracy in terms of KL divergence, cf. Appendix. The invariance of the KL divergence under monotonic transformations, one of its key properties, is then automatically guaranteed.

5 Predictive Discretization

In predictive discretization, 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. For this reason, this leads to a fair assessment of predictive accuracy. Predictive discretization is conceptually slightly more involved than semi-predictive discretization: the finest grid *changes* along the sequence, as it is based on data $D^{(i-1)}$ seen prior to each step i.

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 now exactly one hypercube $[z]_{\Omega_{\Lambda}^{(0)}}$ that is mapped to each x, although there is no data point in any of the hypercubes $[z]_{\Omega_{\Lambda}^{(0)}}$ 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 $k = 1, ..., n$). Again, 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 [12], 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'.[7,8] 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 $\Omega_{\Lambda}^{(i)} = \Omega_{\Lambda'}^{(i)}$.

Despite these conceptual differences to semi-predictive discretization, we obtain an equation that resembles Eq. 7. In the following, we outline each of the three terms regarding predictive discretization. The marginal likelihood $p(D_{\Lambda}|m)$ is the same in both cases (cf. Eq. 8), as it is unaffected by the different mapping from the discretized space to the continuous one.

Regarding the mapping from Z to Y (analogous to the last term in Eq. 7), we obtain the decomposition

$$\left(\prod_{i=i_0+1}^{N} \rho(y^{(i)}|z^{(i)}, \Omega_{\Lambda}^{(i-1)}) \right) \cdot \left(\prod_{i=1}^{i_0} \rho(y^{(i)}|z^{(i)}, \Omega_{\Lambda}^{(i-1)}) \right), \tag{11}$$

which depends on the exact sequential ordering. However, the first term ($i > i_0$) is *identical* regarding both Λ and Λ', and is hence irrelevant when comparing those two discretization policies to each other. Due to the second term in Eq. 11, our predictive scoring function hence depends only on the sequential ordering during a (short) initial phase ($i \leq i_0$). Because of $i_0 \ll N$, the second term in Eq. 11 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. 11 as well.

Let us now consider the mapping from X to Z (analogous to the second term in Eq. 7). Like before, we have to determine the number of hypercubes $[z]_{\Omega_{\Lambda}^{(i-1)}}$ that are mapped to $x^{(i)}$ at each step i. Similarly to Eq. 9, this number

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

[8] 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 $\Lambda^{\cup}$.

is given by

$$\prod_{i=1}^{N}\prod_{k=1}^{n}\frac{1}{N_{+}^{(i-1)}(x_k^{(i)})}=\prod_{k=1}^{n}\prod_{x_k}\frac{1}{\Gamma(N(x_k))}=:G(D,\Lambda)^{-1} \tag{12}$$

where $N^{(i-1)}(\cdot)$ 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 $D^{(i-1)}$. Furthermore, $N_{+}(x_k)=\max\{1,N(x_k)\}$ arises from the fact that, at small $i\leq i_0$, there is at *least one* hypercube mapped to each x, even if it does not contain a data point (yet).[9] Note that 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 8.

Finally, we now obtain the (approximate) predictive scoring function,

$$\mathcal{L}_{\mathrm{P}}(\Lambda,m)=\log p(D_{\Lambda}|m)-\log G(D,\Lambda), \tag{13}$$

which is independent of the specific sequential ordering chosen for each *pair* of discretization policies Λ and Λ'. Moreover, Eq. 13 represents 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.

As a consequence, this property allows us to choose Λ' as the discretization policy that assigns exactly one state to each variable; this resolves the restriction noted in footnote 8, and we can hence assign a score to any discretization policy Λ that does not give rise to an "empty state". 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).

Obviously, the predictive scoring function in Eq. 13 is very similar to the semi-predictive one (cf. Eq. 10). While the previous discussion concerning all the (qualitative) properties of the semi-predictive scoring function carries over, there are also (minor) quantitative differences between our semi-predictive and predictive scoring function. While the marginal likelihood of the graph is identical in both Eq. 10 and 13, the terms penalizing small numbers of discretization levels differ. However, given a sufficiently large data set ($N(x_k)\gg 1$ for all x_k), this difference reads (Stirling approximation):

$$\begin{aligned}&\log p(D_{\Lambda}|\hat{\theta},m_{\mathrm{empty}})-\log G(D,\Lambda) \\ &\approx Nn\log N+\frac{1}{2}\sum_{k=1}^{n}\sum_{x_k}\log N(x_k)+\mathcal{O}(\mathrm{const})\end{aligned} \tag{14}$$

The leading-order term that *depends* on the discretization is $\mathcal{O}(\log N)$. As this is the same order as the one of the complexity penalty implicitly present in

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

$\log p(D_\Lambda|m)$, the difference between semi-predictive and predictive discretization is relevant. Eq. 14 also reveals that the *predictive* scoring function favours (slightly) *more* discretization levels compared to semi-predictive discretization. This follows immediately by induction over the number of discretization levels: consider a discretization level x_k being split into two new levels x'_k and x''_k; regarding the counts, we thus have $N(x'_k) + N(x''_k) = N(x_k)$. Then the new difference is *larger* than the old one (cf. Eq. 14): $\log(N(x'_k)) + \log(N(x''_k)) \geq \log(N(x_k))$ if $N(x'_k), N(x''_k) \geq 2$; and strictly if $N(x'_k), N(x''_k) \geq 3$. The semi-predictive scoring function hence penalizes large numbers of discretization levels more severely, and hence favours fewer discretization levels.

6 Preliminary Experiments

In computational biology, regulatory networks are often modeled by Bayesian networks, and their structures are learned from discretized gene-expression data, see, e.g., [6, 11, 7]. Obviously, one would like to recover the "true" network structure underlying the continuous data, rather than a degraded network structure due to a suboptimal discretization policy. Typically, the expression levels have been discretized in a preprocessing step, rather than jointly with the network structure, [6, 11, 7]. In our experiment, we employed our predictive scoring function (cf. Eq. 13) and re-analyzed the gene expression data concerning the pheromone response pathway in yeast [7], comprising 320 measurements concerning 32 continuous variables (genes) as well as the mating type (binary variable). Based on an error model concerning the micro-array measurements, a continuously differentiable, monotonic transformation is typically applied to the raw gene expression data in a preprocessing step. Since our (semi-)predictive scoring function is invariant under this kind of transformation, this has no impact on our analysis, so that we are able to work directly with the raw data.

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 (semi-)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.

The marginal likelihood $\rho(D_\Lambda|m)$, 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 [13], its value has a decisive impact on the resulting number of edges in the network, and must hence be chosen with great care. Assessing predictive accuracy by means of 5-fold cross validation (cf. Appendix), we determined $\alpha \approx 25$.

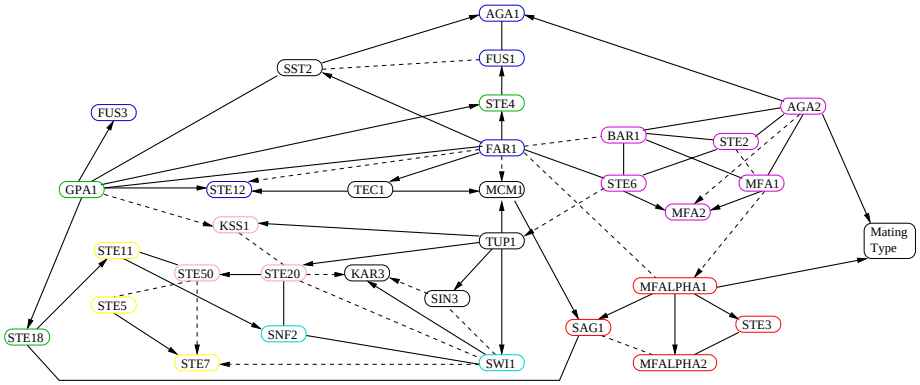

Figure 3: This graph is compiled from 320 delete-30 jackknife samples (cf. [7] for the color-coding).

Fig. 3 shows the composite graph we learned from the used gene expression data, employing our predictive scoring function, cf. Eq. 13.[10] This graph is compiled by averaging over several Bayesian network structures in order to account for model uncertainty prevailing in the small data set. Instead of exploring model uncertainty by means of Markov Chain Monte Carlo in the model space, we used a non-parametric re-sampling method, as the latter is independent of any model assumptions. While the bootstrap has been used in [5, 4, 6, 11], we prefer the jackknife when learning the graph structure, i.e., conditional independences. The reason is that bootstrap samples can contain *multiple* copies of identical data points, which in turn imply strong statistical dependences among the variables when given a small data set D. As a consequence, the resulting network structure can be considerably biased towards denser graphs. The jackknife avoids this problem. We obtained very similar results using three different variants of the jackknife: delete-1, delete-30, and delete-64. Averaging over 320 delete-30 jackknife sub-samples, we found 65.7 ± 8 edges. Fig. 3 displays 65 edges: 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. Apart from that, our predictive scoring function yielded that most of the variables have about 4 discretization levels (on average over the 320 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.

In Fig. 3, it is apparent that the genes AGA2, BAR1, MFA1, MFA2, STE2,

[10] We imposed no constraints on the network structure in Fig. 3. Unfortunately, the results we obtained when imposing constraints derived from location data have to be skipped due to lack of space.

and STE6 (magenta) are densely interconnected, and so is the group of genes MFALPHA1, MFALPHA2, SAG1 and STE3 (red). Moreover, both of those groups are directly connected to the mating type, while the other genes in the network are (marginally) independent of the mating type. This makes sense from a biological perspective, as the former genes (magenta) are only expressed in yeast cells of mating type A, while the latter ones (red) are only expressed in mating type ALPHA; the expression level of the other genes is rather unaffected by the mating type. Due to lack of space, a more detailed (biological) discussion has to be omitted here.

The crucial impact of the used discretization policy Λ and scale-parameter α on the resulting network structure becomes apparent when our results are compared to the ones 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 Fig. 3, 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 [13], this suggests that the major differences in the obtained network structures are actually due to the discretization policy.

7 Conclusions

We have shown that the discretization method can substantially impact the resulting graph structure. This highlights the importance of principled yet efficient methods for finding the resolution at which to represent continuous observations. Our discretization approach 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 (semi-)predictive scoring functions are both simple and computationally efficient.

Acknowledgements

We would like to thank Alexander Hartemink for making the pheromone data available to us. Harald Steck acknowledges support from the German Research Foundation (DFG) under grant STE 1045/1-2. Tommi Jaakkola acknowledges support from Nippon Telegraph and Telephone Corporation, NSF ITR grant IIS-0085836, and from the Sloan Foundation in the form of the Sloan Research Fellowship.

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. *Journal of the Royal Statistical Society, Series A*, 147:277–305, 1984.

[3] N. Friedman and M. Goldszmidt. Discretization of continuous attributes while learning Bayesian networks. In *Proceedings of the International Conference on Machine Learning*, pages 157–65. Morgan Kaufmann, 1996.

[4] N. Friedman, M. Goldszmidt, and A. Wyner. Data analysis with Bayesian networks: A bootstrap approach. In K. Laskey and H. Prade, editors, *Proceedings of the 15^{th} Conference on Uncertainty in Artificial Intelligence*, pages 196–205. Morgan Kaufmann, 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. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, pages 197–202, 1999.

[6] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 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. In *Pacific Symposium on Biocomputing*, 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] S. Monti and G. F. Cooper. A multivariate discretization method for learning Bayesian networks from mixed data. In G. F. Cooper and S. Moral, editors, *Proceedings of the 14^{th} Conference on Uncertainty in Artificial Intelligence*, pages 404–13. Morgan Kaufmann, 1998.

[10] S. Monti and G. F. Cooper. A latent variable model for multivariate discretization. In *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics*, pages 249–54, 1999.

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

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

[13] H. Steck and T. S. Jaakkola. On the Dirichlet prior and Bayesian regularization. In *Advanves in Neural Information Processing Systems (NIPS) 15*, 2002.

[14] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, Series B*, 36:111–47, 1974.

Appendix: Cross Validation

In cross validation [14], data D is split into training data D^{T} and validation data D^{V}. After graph m and discretization policy Λ are learned from the training data, the KL divergence can be used to measure predictive accuracy by comparing the density ρ^{T} predicted by m and Λ to the density ρ^{V} implied by validation data D^{V}. Using the finest grid implied by the (entire) *training* data, represented by Ω^{T}, for *both*[11] ρ^{T} and ρ^{V}, we obtain

$$\rho^{\mathrm{T}}(y|m,\Lambda,D^{\mathrm{T}},\Omega^{\mathrm{T}}) = \frac{\rho(y|z,\Omega^{\mathrm{T}})}{\prod_{k=1}^{n} N_{\Lambda}^{\mathrm{T}}(x_k)}\; p(x|m,\Lambda,D^{\mathrm{T}}) \tag{15}$$

where $x = f_{\Lambda}(y)$ and $z = f_{\Omega^{\mathrm{T}}}(y)$; $N_{\Lambda}^{\mathrm{T}}(\cdot)$ are the counts due to the training data D^{T} discretized according to Λ; like before, we allow for an arbitrary, but strictly positive $\rho(y|z,\Omega^{\mathrm{T}}) = \prod_{k=1}^{n} \rho(y_k|z_k,\Omega_k^{\mathrm{T}})$; and

$$\rho^{\mathrm{V}}(y|D^{\mathrm{V}},\Omega^{\mathrm{T}}) = \frac{N_{\Omega^{\mathrm{T}}}^{\mathrm{V}}(z)}{N^{\mathrm{V}}}\; \rho(y|z,\Omega^{\mathrm{T}}) \tag{16}$$

where $z = f_{\Omega^{\mathrm{T}}}(y)$; $N_{\Omega^{\mathrm{T}}}^{\mathrm{V}}(\cdot)$ are the counts implied by the *validation* data D^{V} w.r.t. the finest grid implied by the *training* data, Ω^{T}; $\rho(y|z,\Omega^{\mathrm{T}})$ is identical to the one in Eq. 15. This yields immediately

$$\begin{aligned}
\mathrm{KL}(\rho^{\mathrm{V}}||\rho^{\mathrm{T}}) &= \int dy\; \rho^{\mathrm{V}} \log \frac{\rho^{\mathrm{V}}}{\rho^{\mathrm{T}}} \\
&= -\sum_{x} \hat{p}(x|\Lambda,D^{\mathrm{V}}) \log \frac{p(x|m,\Lambda,D^{\mathrm{T}})}{p(x|\hat{\theta},m_{\mathrm{empty}},\Lambda,D^{\mathrm{T}})} + \tilde{c}
\end{aligned} \tag{17}$$

where $\hat{p}(x|\Lambda,D^{\mathrm{V}}) = N_{\Lambda}^{\mathrm{V}}(x)/N^{\mathrm{V}}$; $p(x|\hat{\theta},m_{\mathrm{empty}},\Lambda,D^{\mathrm{T}}) = \prod_{k=1}^{n} N_{\Lambda}^{\mathrm{T}}(x_k)/N^{\mathrm{T}}$; $N_{\Lambda}^{\mathrm{V}}(\cdot)$ and $N_{\Lambda}^{\mathrm{T}}(\cdot)$ are the counts implied by D^{V} and D^{T}, respectively, discretized according to Λ. The constant $\tilde{c}$ is independent of Λ and m, and hence irrelevant when comparing different discretization policies and graphs in light of the same data. The KL divergence depends only on the counts in the discretized domain, like our (semi-)predictive scoring functions (cf. Eqs. 10 and 13). It is hence independent of the metric in the continuous space and invariant under monotonic transformations of the variables. The KL divergence in Eq. 17 is a weighted sum of the log ratios involving the regularized graphical model and the unregularized empty graph model, which is very similar to Eq. 10. While this ratio is based on the training data, the weights of the sum are determined by the validation data.

[11] Again, the same grid is used for computational efficiency.