(Semi-)Predictive Discretization During Model Selection

Arto Klami

28.4.2004

1 Introduction

In this paper, I summarize the article "(Semi-)Predictive Discretization During Model Selection" by Harald Steck and Tommi S. Jaakkola [6]. The article deals with data discretization — more specifically it introduces a method that can be used to select optimal discretization while selecting the structure of a graphical model.

1.1 Discretization

Many data sets in machine learning are given as continuous features, i.e. they consist of vectors of arbitrary real values. Discretization means a mapping from the continuous values into a set of discrete values. Here only *deterministic discretizations* are considered, i.e. for each real value we have exactly one discrete value.

Data discretization is needed for various reasons. One reason is that there are many machine learning algorithms that can only be applied to discrete data. In order to use those algorithms, we need to discretize the data. We might also want to do that for solely computational reasons; some problems are easier to compute for discrete variables. Finally, if we know that our data is discrete, but we only have noisy continuous measurements, we would naturally want to discretize the data to correspond with the underlying discrete values.

The discretization is usually done as a preprocessing step, before the actual analysis. In theory, we can use any clustering algorithm for data discretization, but there is no reason why such methods would provide an optimal discretization. Several methods have been developed to create, in some sense, optimal discretization for multivariate data sets. The basic idea behind most such methods is to try to preserve the information of other variables provided by one. For example, [1] proposed a method where the goal is to retain the class entropy, and [5] gave similar approach for unlabeled data sets.

1.2 Graphical models

Graphical model means a Bayesian network that models a set of variables by a graph of dependencies and conditional probabilities. Given N samples of each of the n variables, the task is to find the structure of the graph (denoted by m). Many methods have been suggested for the problem of finding the correct model structure in the case of discrete variables. There are also some methods for special cases of continuous variables (e.g all distributions are Gaussian).

2 Discretization During Model Selection

The article proposes a method for discretizing the data during the model selection, instead of doint it as a preprocessing step. The idea is applied to graphical models, and the task is to learn the graph structure and the discretization at the same time. A few methods have earlier been suggested for the same task [2, 4], but the authors claim that the proposed methods are computationally too heavy to be used in practice.

The n continuous variables are here denoted by $Y = (Y_1, ..., Y_n)$, and the *discretization policy* by $\Lambda = (\Lambda_1, ..., \Lambda_k)$. Each variable is discretized according to a sequence of *threshold values* $\Lambda_k = (\lambda_{k,1}, ..., \lambda_{k,r_k-1})$, where r_k is the number of discretization levels for kth variable. The discretized variables are denoted by $X = (X_1, ..., X_n)$, and they are obtained from Y by the mappings $f_\Lambda : Y \rightarrow X$. The discretized values x_k are then given by $f_{\Lambda_k}(y_k)$, which is j for $\lambda_{k,j-1} \leq y_k < \lambda_{k,j}$.

2.1 Sequential Approach

More formally, the task is to maximize the likelihood of observed continuous data D given the discretization policy Λ and the model structure m. The likelihood is

here computed in a sequential manner,

$$p(D|\Lambda, m) = \prod_{i=1}^{N} p(y^{(i)}|D^{(i-1)}, \Lambda, m) \quad (1)$$

where $D^{(i-1)} = (y^{(i-1)}, ..., y^{(1)})$ denotes the data points seen prior to step i along the sequence.

Because the discretization is deterministic, we can factor the predictive distribution as

$$p(y^{(i)}|D^{(i-1)}, \Lambda, m) = p(y^{(i)}|x^{(i)}, \Lambda)p(x^{(i)}|D^{(i-1)}, m, \Lambda) \; . \quad (2)$$

Assuming the dependences among the continuous variables Y_k are described by the underlying discretized distribution $p(X|D, \Lambda, m)$, the continuous variables Y are independent given X,

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

The second term in (2) denotes the predictive distribution of a discrete variable given the previous discrete variables. As stated earlier, various methods for optimizing the model structure exist for discrete data, and we could use them is this was the only term in the cost. However, we also have the first term that represents the distribution of continuous values given the discretized values, and we need to study that further.

2.2 The Finest Grid

In order to keep the problematic term computable, we need a concept called finest grid. By that we mean (for each variable) a discretization sequence Ω_k that discretizes the data set such that there is exactly one data point in each discretization level. In other words, there is one threshold between any two closest data samples, and we can choose whichever point we want.

We restrict our actual discretization policy Λ so that the threshold values are chosen from the set of threshold values of the finest grid. Note that this is not actually a restriction, because we were able to select the threshold values of Ω freely between the data points.

If we denote by Z the discretized value according to the finest grid, we can factor the distribution of continuous values given the discrete ones by first mapping X to Z and then Z to Y. This gives

$$p(y_k^{(i)}|x^{(i)}, \Lambda_k, \Omega_k) = p(y_k^{(i)}|z_k^{(i)}, \Omega_k)p(z_k^{(i)}|x^{(i)}, \Lambda_k, \Omega_k) \; , \quad (4)$$

where everything is now conditional on the particular finest grid we chose.

Here the authors make one assumption, namely that the probability mass predicted for $x^{(i)}$ is divided evenly among the cells $z^{(i)}$ of the finest grid that are mapped to the particular x. That is,

$$p(z_k^{(i)}|x^{(i)}, \Lambda_k, \Omega_k) = \frac{1}{N(x_k^{(i)})} \; , \quad (5)$$

where $N(x_k^{(i)})$ is the number of data points in the discretization level $x_k^{(i)}$.

3 Semi-Predictive Discretization

Given the previous two assumptions leading to (3) and (5), we can write the likelihood (1) in a form that makes computation possible. However, there is a slight problem. The likelihood is derived in the sequential manner, and every prediction is made only based on previous samples. Unfortunately, we need the finest grid in order to compute the predictions, and it is based on the whole data set. In this sense, the cost is not fully predictive, and therefore the resulting method is called semi-predictive discretization.

Plugging (2), (3), (4) and (5) into (1), we finally get

$$p(D|\Lambda, m, \Omega) = p(D_\Lambda|m) \left(\prod_{i=1}^{N} \prod_{i=1}^{N} \frac{1}{N(x_k^{(i)})} \right) \left(\prod_{i=1}^{N} \prod_{i=1}^{N} p(y_k^{(i)}|z_k^{(i)}, \Omega_k) \right) \; . \quad (6)$$

This likelihood consists of three terms that are studied next.

The first term is the likelihood given discretized data. This can be maximized with respect to m relatively easily, as everything is discrete. Such methods are not within the scope of this summary.

The next term is because of the mapping from X to Z, and is computable because of the assumption of even distribution. We can also write the term (times constant) as the reciprocal of the maximum likelihood of an empty graph (all variables are independent). Denote that by $p(D_\Lambda|\hat{\theta}, m_{empty})$, where $\hat{\theta}$ is the maximum likelihood estimate of the parameters $\hat{\theta}_{x_k} = N(x_k)/N$.

The third term denotes the mapping from Z to Y. It is the only term in the likelihood that depends on the

metric of the original continuous data space and the particular finest grid we chose. However, it does not depend on Λ or m, and thus it is irrelevant when we are comparing discretizations or model structures.

By dropping the last term and taking a logarithm, we get the cost function of semi-predictive discretization:

$$\begin{aligned} l_{SP}(\Lambda, m) &= \log p(D_\Lambda|m) - \log p(D_\Lambda|\hat{\theta}, m_{empty}) \\ &= \log p(D_\Lambda|m) + N \sum_{k=1}^{n} H(\hat{p}(X_k)) \ . \end{aligned} \quad (7)$$

On the second line, the likelihood of the empty graph is written as the entropies of the discrete distributions. The first term measures the ability to predict the discrete variables, and it is naturally easier if we have fewer discretization levels (with one level we get always correct predictions). This is compensated by the second term, which penalizes from too few discretization levels.

The cost has a few interesting properties. First, it depends only on the counts of samples at different discretization levels, which makes computations simple. Second, it is independent of the particular choice of the finest grid, which justifies the arbitrary selection of the thresholds. Third, it is independent of the metric in the continuous space, rendering the use of preprocessing methods unnecessary.

4 Predictive Discretization

The semi-predictive discretization method was not fully justified, because the finest grid was based on the whole data. Fortunately, that is not really necessary. We can derive a similar cost along the same lines by allowing the finest grid to adapt with new data samples. The details can be found in the original paper, and the resulting cost function is

$$l_P(\Lambda, m) = \log p(D_\Lambda|m) - \log G(D, \Lambda) \ ,$$

where

$$G(D, \Lambda) = \left(\prod_{k=1}^{n} \prod_{x_k} \frac{1}{\Gamma(N(x_k^{(i)}))} \right)^{-1} \ .$$

The only difference in the cost functions is in the penalty term. It can be shown that the predictive distribution penalizes small numbers of discretization levels slightly more, and thus favors somewhat finer discretization. It is not stated in the paper explicitly, but it seems we should always use the predictive discretization method, because it is theoretically valid. The semi-predictive discretization is retained because it is easier to present and understand.

5 Empirical Experiments

The predictive discretization method is briefly tested in the paper in one application. The data is a gene expression data concerning the pheromone response pathway in yeast, and it consists of 320 measurements of 32 continuous variables (genes) and one binary variable (mating type). The same data has been analyzed earlier [3] with discretization as a preprocessing step, and the resulting model structure resembled closely a naive-Bayes network with the mating type as a root variable and other variables pretty much independent from each other.

The network structure obtained with predictive discretization is completely different. There are two clear groups of variables that are strongly interconnected, and according to the authors these correspond to two different mating types. In other words, the model structure seems to be biologically plausible, but further analysis would be required to say whether it is in some sense correct or not.

6 Conclusions

The main conclusion to be drawn from the paper (especially the experiment section) is that the discretization drastically affects the resulting model structure. Therefore we cannot just select some discretization method, but we need to really try to find a discretization that preserves the dependencies. This makes discretization an important field of study.

The paper also has some good ideas that can possibly be used in other kinds of model learning problems. The concept of finest grid made here computations relatively simple by making the cost function depend only on the counts of data samples. The finest grid could be used in other situations also, or we could use other tricks to transform the cost to depend only on the counts. If the cost depends on the actual partition on the data samples into the discretization levels, the number of possible discretizations is exponential in the number of samples. Dependence on only the counts reduces this significantly.

Another interesting observation is the form of the final cost function. It consists of the likelihood of the discrete data and a separate penalty term for too coarse discretizations. Searching for such form of cost function in other approaches could be beneficial, because the part for discrete data is often relatively easy to compute in cases where we want to discretize data.

References

[1] U.M. Fayyad and K.Irani. Multi-interval discretizaion of continuous-valued attributes for classification learning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1022–1027, 1993.

[2] N. Friedman and M. Goldszmidt. Discretization of continuous attributes while learning bayesian networks. In *Proceedings of the International Conference on Machine Learning*, pages 157–165. Morgan Kaufmann, 1996.

[3] A.J. Hartemink, D.K. Gifford, T.S. Jaakkola, and R.A. Yound. Combining location and expression data for principled discovery of genetic regulatory networks. In *Pacific Symposium on Biocomputing*, 2002.

[4] S. Monti and G.F. Cooper. A multivariate discretization methods for learning bayesian networks from mixed data. In G.F. Cooper and S. Moral, editors, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 404–413. Morgan Kaufmann, 1998.

[5] 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 249–254, 1999.

[6] Harald Steck and Tommi S. Jaakkola. (semi-)predictive discretization during model selection. AI Memo AIM-2003-002, 2003.