An Efficient Search Strategy for Aggregation and Discretization of Attributes of Bayesian Networks Using Minimum Description Length

by

Dai (Daniel) Tran

B.S. in Civil Engineering, National University of Civil Engineering, 2003

M.S. in Structural Engineering, Georgia Institute of Technology, 2009

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Master of Science

Department of Applied Mathematics

2013

This thesis entitled:
An Efficient Search Strategy for Aggregation and Discretization of Attributes of Bayesian Networks Using Minimum Description Length
written by Dai (Daniel) Tran
has been approved for the Department of Applied Mathematics

Jem N. Corcoran

Dr. Keith R. Molenaar

Date

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Tran, Dai (Daniel) (M.S., Applied Mathematics)

An Efficient Search Strategy for Aggregation and Discretization of Attributes of Bayesian Networks Using Minimum Description Length

Thesis directed by Dr. Jem N. Corcoran

Bayesian networks are widely considered as powerful tools for modeling risk assessment, uncertainty, and decision making. They have been extensively employed to develop decision support systems in variety of domains including medical diagnosis, risk assessment and management, human cognition, industrial process and procurement, pavement and bridge management, and system reliability. Bayesian networks are convenient graphical expressions for high dimensional probability distributions which are used to represent complex relationships between a large number of random variables. A Bayesian network is a directed acyclic graph consisting of nodes which represent random variables and arrows which correspond to probabilistic dependencies between them.

The ability to recover Bayesian network structures from data is critical to enhance their application in modeling real-world phenomena. Many research efforts have been done on this topic to identify the specific network structure. However, most Bayesian network learning procedures are based on the following two assumptions: (1) that the data are discrete **or** (2) that the data are continuous and either follow a Gaussian distribution or are otherwise discretized before recovery. Discretization of data in the continuous non-Gaussian case is often done in an ad hoc manner which destroys the conditional relationships among variables– subsequent network recovery algorithms are then unable to retrieve the correct network. Friedman and Goldszmidt [11] suggest an approach based on the minimum description length principle that chooses a discretization which preserves the information in the original data set, however it is one which is difficult, if not impossible, to implement for even moderately sized networks. This thesis explores a structure of the minimum description length developed and then provides an alternative efficient search strategy which allows one to use the Friedman and Goldszmidt in practice.

Acknowledgements

I am greatly indebted to my advisor, Dr. Jem Corcoran, for her patience, encouragement, and confidence in me throughout this study. Without her guidance and continued support, this study would have not been possible. Jem's superior intellect, attention to detail, and supportive nature make her the best of my mathematics instructors. Thank you so much for inspiring and bringing me into the stochastic world.

I would like to express sincere gratitude to Dr. Keith Molenaar and Dr. Ray Littejohn for providing me with insightful advice and guidance on several aspects of this study.

I thank the University of Colorado, Department of Civil Environmental and Architectural Engineering for their financial support of this study.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

INTRODUCTION

1.1 Introduction

Modeling real-world phenomena for decision making is often complex due to risk and uncertainty. Artificial intelligence researchers typically use a knowledge-based approach while statisticians traditionally employ a data-based approach towards obtaining this goal. Bayesian networks, which can combine historical data and expert judgment, provide a vehicle to rigorously analyze and quantify risk relevant to the decision. Bayesian networks modeling is widely consider as a powerful technique for handling risk assessment, uncertainty, and decision making [10]. Bayesian Networks have been extensively employed to develop decision support systems in variety of domains including medical diagnosis, risk assessment and management, human cognition, industrial process and procurement, pavement and bridge management, and system reliability.

Bayesian networks are convenient graphical expressions for high dimensional probability distributions, representing complex relationships between a large number of random variables(Pearl [22]). A Bayesian network (BN) is a **directed acyclic graph** consisting of nodes which represent random variables and arrows which correspond to probabilistic dependencies between them. As described in Chapter 2, the "parent-child" structure encoded in the graph succinctly describes a set of **conditional independence assertions** which allows one to distinguish between the idea of correlation of random variables and the more subtle notion of direct dependence.

In the late 1980's and early 1990's, the popularity of Bayesian networks surged as their usefulness in encoding uncertain expert knowledge in expert systems was realized and developed

[15]. Specifically, they afford great flexibility for incorporating prior knowledge into data analysis. Additionally, they are useful for handling incomplete data sets and allowing researchers to learn about causal relationships between variables.

There has been a great deal of work done ([5], [8], [12], [13], [14], [16], [18], [19], [21]) on the problem of recovering (learning) the structure of a generating network from data. The majority of Bayesian network recovery methods studied in the literature to date apply to networks made up of nodes representing discrete random variables, or, in the continuous case, the assumption is that the random variables are Gaussian. For general continuous data, network recovery typically begins by either first discretizing the data or proceeding as if the Gaussian assumption is correct. Often ([9], [17], [19], [26]) this discretization is performed in an ad hoc manner. Unfortunately, such a non-rigorous approach is highly likely to destroy the precise conditional dependencies one is out to recover. Friedman and Goldszmidt [11] suggest an approach based on **minimum description length principle** that chooses a discretization which preserves the information in the original data set. However, this approach is challenging, and in some cases impossible, to implement for even moderately sized networks. In this thesis, we provide an extremely effective search strategy which allows one to use the Friedman and Goldszmidt approach in practice.

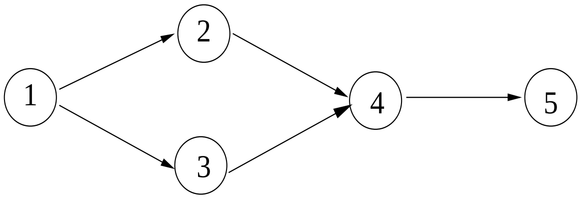
1.2 Bayesian Networks

Probabilistic graphical models are graphs in which nodes represent random variables, and arcs (lack of arcs) represent conditional dependence (independence) assumptions. A directed acyclic graph (DAG) is a set of nodes and directed arcs which do not form a closed loop or cycle. Consider the directed acyclic graph on five nodes shown in Figure 1.1.

Figure 1.1 indicates that node 1 is a **parent** of nodes 2 and 3 and nodes 2 and 3 are **children** of node 1. In general, consider a DAG with n nodes. For $i = 1, 2, \ldots, n$, let Π_i denote the set of parents of node i. For the graph in Figure 1.1, we have

$$\Pi_1 = \emptyset, \qquad \Pi_2 = \{1\}, \qquad \Pi_3 = \{1\}, \qquad \Pi_4 = \{2, 3\}, \qquad \text{and} \quad \Pi_5 = \{4\}.$$

Figure 1.1: A Directed Acyclic Graph



A **Bayesian network** consists of a DAG and a set of conditional probability distributions $P(X_i|\Pi_i)$, for $i = 1, 2, \ldots n$ along with the assumption that the joint probability density function for all n nodes in the network is given by

$$p(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(x_i|\Pi_i). \tag{1.1}$$

where $p(x_i|\Pi_i)$ is a specified conditional probability distribution for the random variable X_i given the values of it parent random variables.

For example, if the DAG shown in Figure 1.1 represents a Bayesian network, then the joint density $p(x_1, x_2, x_3, x_4, x_5)$ for the random variables X_1, X_2, X_3, X_4, and X_5 is given by

$$p(x_1, x_2, x_3, x_4, x_5) = p(x_1)\, p(x_2|x_1)\, p(x_3|x_1)\, p(x_4|x_2, x_3)\, p(x_5|x_4)$$

It should be noted that in this DAG that, once the value of X_1 is fixed, the random variables X_2 and X_3 are independent. That is, X_2 and X_3 are conditionally independent given the value of X_1. Similarly, the random variables X_1 and X_4 are independent once the value of X_2 is given. For a general Bayesian network, the following observations can be seen based on the assumption given by (1.1):

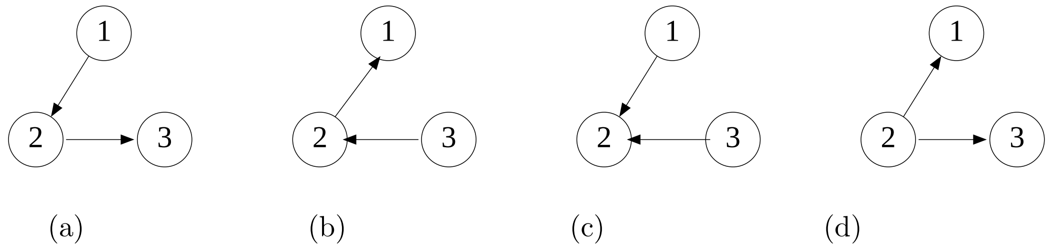
- children of common parents are conditionally independent given their parents,

  and more generally,

- each random variable X_i in a Bayesian network is independent of it's non-descendants, given it's parents.

To further illustrate this point, Figure 1.2 presents four DAGs containing the same edges, ignoring directions. The four joint probability densities associated with these four DAGS are shown below.

Figure 1.2: Four DAGs With the Same (Undirected) Edges



(a) (b) (c) (d)

$$
\begin{aligned}
p(x_1, x_2, x_3) &= p(x_1)p(x_2|x_1)p(x_3|x_2) && \text{(a)} \\
p(x_1, x_2, x_3) &= p(x_3)p(x_2|x_3)p(x_1|x_2) && \text{(b)} \\
p(x_1, x_2, x_3) &= p(x_1)p(x_3)p(x_2|x_1, x_3) && \text{(c)} \\
p(x_1, x_2, x_3) &= p(x_2)p(x_1|x_2)p(x_3|x_2) && \text{(d)}
\end{aligned}
\tag{1.2}
$$

From DAGs (a), (b), and (d) in (1.2) and Figure 1.2, one can observe that

- X_1 and X_2 are dependent,
- X_2 and X_3 are dependent,
- and therefore X_1 and X_3 are dependent,
- however X_1 and X_3 are independent given X_2.

In contrast, DAG (c) is oriented such that

- X_1 and X_2 are dependent,
- X_2 and X_3 are dependent,
- but X_1 and X_3 are independent.

A set of DAGs are said to be **Markov equivalent** if these DAGs share the same set of conditional independence relationship among variables. For example, DAGs (a), (b), and (d) in Figure 1.2 are **Markov equivalent** because these four graphs encode the same dependencies. It should be noted that network recovery algorithms run on a fixed data set that can not distinguish between Markov equivalent graphs. The distinction can be made if one has the ability to generate or collect data where certain nodes are being held to fixed values. In this thesis, we will focus only on graph recovery up to the *Markov equivalence class*.

1.3 Thesis Organization

This thesis is organized as follows. Chapter 2 presents an overview of recovering of Bayesian networks including multinomial networks, network scores, network recovery, and discretizing data. In Chapter 3, we discuss the general principle of minimum description length (MDL) for Bayesian networks and Friedman and Goldszmidt's adaptation of MDL for the purpose of discretization of data. In Chapter 4 we introduce an effective search strategy that can be used to optimize the search for Friedman and Goldszmidt's discretization score. Finally, Chapter 5 discusses the findings, limitations and future work from this study.

Chapter 2

OVERVIEW OF RECOVERING BAYESIAN NETWORKS

2.1 Introduction

The recovery of Bayesian network structure from data is critical to enhance their application in modeling real-world phenomena. Many research efforts have been done on this topic to identify the specific network structure ([5],[8],[12], [13], [14],[16],[18],[19], and [21]). However, most Bayesian network learning procedures are based on the following two assumptions: (1) that the data are discrete **or** (2) that the data are continuous and either follow a Gaussian distribution or are otherwise discretized before recovery.

For discrete data, the most common distributional assumption made about Bayesian networks is that the nodes (random variables) have a multinomial distribution. The multinomial distribution is an extension of the binomial distribution to the case of where there are more than two classes into which the outcome of an experiment can fall.

For continuous data, a very popular assumption for Bayesian networks with nodes representing continuous variables is that all the random variables follow Gaussian distributions, and the relationships between variables are linear. This form of network is usually called a *Gaussian belief network*. Often, recovery tools specific to Gaussian belief networks are used in the case of continuous data, even when their use is not warranted. Sometimes, ignoring the validity of certain assumptions in statistical models, one can still get reasonable results. However, Wang ([25]) points out that failure of testing both assumptions for multivariate normal distribution and linear dependent association between nodes can lead to an inappropriate use of the Gaussian belief network.

For continuous data that does not appear to be Gaussian, it is usual in the literature to discretize data before recovery. However, in many cases the data is discretized in an ad hoc manner which often destroys the conditional relationships among variables. As discretization of continuous random variables involves reassigning all values in a particular interval to a single value, it is, in a sense, a classification problem which may also be encountered in problems with originally discrete data. This Chapter briefly summarizes formalities and previous work on (1) multinomial networks; (2) network score; (3) network recovery; and (4) discretizing data.

2.2 Multinomial Networks

The most common distributional assumption made about Bayesian networks in the case that the nodes represent **discrete random variables** is that these random variables have a multinomial distribution. This section describes the connection between a Bayesian network and the multinomial distribution.

2.2.1 The Multinomial Distribution

Consider an experiment with m independent trials and r possible outcomes on each trial. Let θ_i, for $i = 1, 2, \ldots, r$ be the probability that any one trial results in outcome i. Define the random variables $X_1, X_2, \ldots, X_r$ where X_i is the number of trials that result in outcome i. Then the vector $(X_1, X_2, \ldots, X_r)$ has a **multinomial distribution** with parameters m, r, and $\theta = (\theta_1, \theta_2, \ldots, \theta_r)$.

The probability mass function for $X = (X_1, X_2, \ldots, X_r)$ is given by

$$P(X_1 = x_1, X_2 = x_2, \ldots, X_r = x_r) = \frac{m!}{x_1! x_2! \cdots x_r!} \theta_1^{x_1} \theta_2^{x_2} \cdots \theta_r^{x_r} \tag{2.1}$$

where $\sum_{i=1}^{r} \theta_i = 1$ and $x_1, x_2, \ldots, x_r$ are non-negative integers summing to m.

2.2.2 The Connection between Bayesian Networks and Multinomial Distributions

Consider now a Bayesian network on n nodes. Let X_i, for $i = 1, 2, \ldots, n$ denote the random variable associated with node i. We assume that X_i can take on r_i values and, for simplicity, we will assume that the r_i values are the integers $1, 2, \ldots, r_i$. We express the dependency of each variable X_i on its parents as:

$$p(X_i = k|\Pi_i) = \theta_{i,\Pi_i,k}$$

where Π_i is a particular configuration of the parent variables of X_i. We will also use the reduced subscript notation

$$\theta_{i,k} = p(X_i = k)$$

if $\Pi_i = \emptyset$.

If we enumerate the number of possible configurations of values taken on by the parent nodes of X_i as $1, 2, \ldots, q_i$ where $q_i = |\Pi_i|$, then we may write

$$p(X_i = k|\Pi_i = j) = \theta_{i,j,k} \tag{2.2}$$

for

$$i = 1, 2, \ldots, n, \qquad j = 1, 2, \ldots, q_i, \qquad \text{and} \ \ k = 1, 2, \ldots, r_i.$$

Now consider data consisting of m observations of the n nodes of a network, and restrict attention for a moment to the m values of the ith node. Consider any one configuration j of the parent nodes to node i that exists in the data. Let m_j be the number of times that the parents of node i take on configuration j in the data set. Then within the m_j values of $X_i|\Pi_i = j$, we can describe the number of observed 1's, 2's, and so on, up to the number of r_i's with a multinomial distribution with parameters m_j, r_i, and $(\theta_{i,j,1}, \theta_{i,j,2}, \ldots, \theta_{i,j,r_i})$.

In this way, using (2.1), we may write the likelihood for the entire $m \times n$ data set D as

$$L_D(\theta) = \prod_{i,j,k} \theta_{ijk}^{n_{ijk}} \tag{2.3}$$

where n_{ijk} is the total number of times in the sample that X_i is observed to have value k when it's parents take on configuration j. (The likelihood is any function proportional to the joint pdf for m independent copies of $(X_1, X_2, \ldots, X_n)$ considered as a function of the θ's.)

2.2.3 Priors on Parameters

The terminology *Bayesian network* derives from the application of Bayes rule in order to determine certain conditional probabilities. A study using Bayesian networks does not necessarily imply a Bayesian modeling approach. However, in the case that one wishes to use Bayesian inferential methods, it becomes necessary to assign prior distributions to the network parameters given by θ_{ijk}. Typically, for multinomial networks, one uses the conjugate prior given by the **Dirichlet distribution**. That is, we will assume that the joint density for the θ_{ijk} for a particular Bayesian network BN, is given by

$$p(\theta|BN) = \frac{\Gamma(\sum \alpha_{ijk})}{\prod \alpha_{ijk}} \prod \theta_{ijk}^{\alpha_{ijk}-1}$$

for some fixed hyperparameters $\alpha_{ijk} > 0$. Note that this is a high dimensional generalization of the more familiar Beta distribution. It is a convenient way to assign values between 0 and 1 to each θ_{ijk} in a way such that $\sum_k \theta_{ijk} = 1$. It is called a **conjugate** prior for the multinomial distribution because if the data given the θ_{ijk} follow a multinomial distribution and our "prior" belief about the θ_{ijk} before observing the data is that they follow a Dirichlet distribution, then the the **posterior** joint distribution of the θ_{ijk} given the data (i.e. after we have observed the data) is another (different parameter) Dirichlet distribution. This is a mathematical convenience for Bayesian analysis.

With this Dirichlet prior, the probability, for any particular Bayesian network BN, of us seeing the data set D is

$$\begin{aligned}
p(D|BN) &= \int\int p(D|BN,\theta)\cdot p(\theta|BN)\,d\theta \\
&= \int \prod_{i=1}^{n}\prod_{j=1}^{q_i^*}\theta_{ijk}^{n_{ijk}}\frac{\Gamma\left(\sum_{k=1}^{r_i}\alpha_{ijk}\right)}{\prod_{k=1}^{r_i}\Gamma\left(\alpha_{ijk}\right)}\prod_{k=1}^{r_i}\theta_{ijk}^{\alpha_{ijk}-1}\,d\theta \qquad (2.4) \\
&= \prod_{i=1}^{n}\prod_{j=1}^{q_i^*}\frac{\Gamma\left(\sum_{k=1}^{r_i}\alpha_{ijk}\right)}{\Gamma\left(\sum_{k=1}^{r_i}(\alpha_{ijk}+n_{ijk})\right)}\prod_{k=1}^{r_i}\frac{\Gamma(\alpha_{ijk}+n_{ijk})}{\Gamma(\alpha_{ijk})}
\end{aligned}$$

where q_i^* is the number of distinct configurations of parents of node i observed in the data. This is as opposed to q_i which is the total number of possible configurations of parents of node i, though q_i^* may be replaced by q_i since the lack of parent configuration j in the data will be reflected by n_{ijk} taking on the value 0.

2.3 Network Scores

There are many ways to recover networks from data. Indeed, we may not even want to think in terms of "one best network" and instead use a model averaging approach or one that constructs a best network by combining best "features" (for example edges) from several networks. In this paper, we will restrict our attention to simple methods for recovering a single "best" network as measured by various standard likelihood and information criterion indices. We will assume a manageable number of networks to score. Of course, the number of possible networks increases superexponentially in the number of nodes– the results of this paper might then be applied using Monte Carlo search strategies.

2.3.1 Log-Likelihood

Given m n-tuples of data points, $u_1, u_2, \ldots, u_m$, the likelihood function for a Bayesian network is given by

$$L(\theta) = \prod_{i=1}^{m} p(u_i) = \prod_{i,j,k}\theta_{ijk}^{n_{ijk}},$$

where n_{ijk} is the total number of times in the sample that X_i is observed to have value k when it's parents take on configuration j.

Given m n-tuples of data points, we compute the log-likelihood for every possible graph. For each DAG, we have a different set of relevant θ parameters. Given a particular DAG, we estimate each θ with its maximum likelihood estimator

$$\hat{\theta}_{ijk} = \frac{\#\text{ observations with } X_i = k \text{ and } \Pi_i = j}{\#\text{ observations with } \Pi_i = j},$$

and then we report the log-likelihood

$$\ln(\hat{\theta}) = \sum_{i,j,k} n_{ijk} \ln(\hat{\theta}_{ijk}).$$

In the event that there are no observations where $\Pi_i = j$, we set $\hat{\theta}_{ijk} = 1$. However, it is important to note that we can always increase the likelihood by including additional θ parameters. Therefore, we will observe the greatest likelihoods ("most likely models") to coincide with DAGs with a maximal number of edges. Thus, the log-likelihood alone is not useful for recovering networks. However, it is the building block for other scoring criteria which generally include penalties for overparameterized models. The two most common penalized likelihood statistics are given by the following information criteria.

2.3.2 Akaike's Information Criterion (AIC)

Akaike's Information Criterion (AIC) is essentially a simple transformation of the above defined likelihood with a term included that penalizes for overparameterization. In the most general form, the AIC is defined by:

$$AIC = -2 \ln L + 2 \cdot (\#parameters).$$

Clearly, the goal is to minimize the AIC to ensure a good fitting model in the sense of maximizing the log-likelihood while penalizing for having too many parameters.

2.3.3 Bayesian Information Criterion (BIC)

The Bayesian Information Criterion (BIC) is defined by

$$BIC = -2 \ln L_D(\hat{\theta}) + (\#\text{ parameters}) \cdot \ln(m),$$

where m is, as before, the sample size. As with the AIC, the goal is to minimize the BIC.

Both the AIC and BIC have rigorous justifications, from both Bayesian and frequentist points of view. AIC was derived from information theory though it can be though of as Bayesian if one uses a clever choice of prior. On the other hand BIC, originally derived through Bayesian statistics as a measure of the Bayes factor, can also be derived as a non-Bayesian result. For more information on these widely used scoring criteria, we refer the interested reader to [1],[2], [3], and [4] (AIC), and [20] and [24] (BIC). To make some broad generalizations, though we have experienced much success with the AIC, it can often overfit the model in terms of number of parameters. BIC, on the other hand, tends to overpenalize, or underfit the model.

2.3.4 Minimum Description Length

As an alternative to AIC and BIC, the *minimum description length principle* (MDL principle) states that the best model is the one which allows for the shortest description, in the sense of encoding, of the data and model itself. With its origins in computer science and information theory, “description length” is the number of bits required to store such an encoding. Unlike AIC and BIC, the concept of minimum description length does not seem to be a familiar one to statisticians and mathematicians. Chapters 3 and 4 present the general principle of using MDL and an efficient search strategy for aggregation and discretization of Bayesian networks in detail.

2.4 Network Recovery

In this section we illustrate recovery of a Bayesian network on a simple three node example. In this case we are able to easily evaluate the AIC and BIC scores for all possible DAGs. Since the number of possible DAGs increases super-exponentially as the number of nodes increases (see Table 2.1), evaluating the scoring criteria for every DAG can quickly become overwhelming. In these cases, it may become necessary to implement network space search methods. Examples of search methods are the greedy hill-climbing, stochastic hill climbing, simulated annealing, and Markov Chain Monte Carlo (MCMC).

Table 2.1: Number of Nodes Versus Number of DAGs

3 nodes	25 dags
4 nodes	543 dags
5 nodes	29,281 dags
6 nodes	3,781,503 dags
7 nodes	1,138,779,265 dags
8 nodes	783,702,329,343 dags

Given a three node network with three random variables our goal is to recover the arrows (edges) that describe their joint probability distribution. The list of all 25 DAGs corresponding to 3 node networks can be found in Table 2.2 and we will refer to these DAGs as they are numbered here throughout this thesis.

We will assume that the data associated with each node is multinomial and they take on the values 1,2,3,4,5,6. In our previously used notation, this means r_1=r_2=r_3=6. In order to simulate data from network 8, Table 2.2 we need to specify the following probabilities.

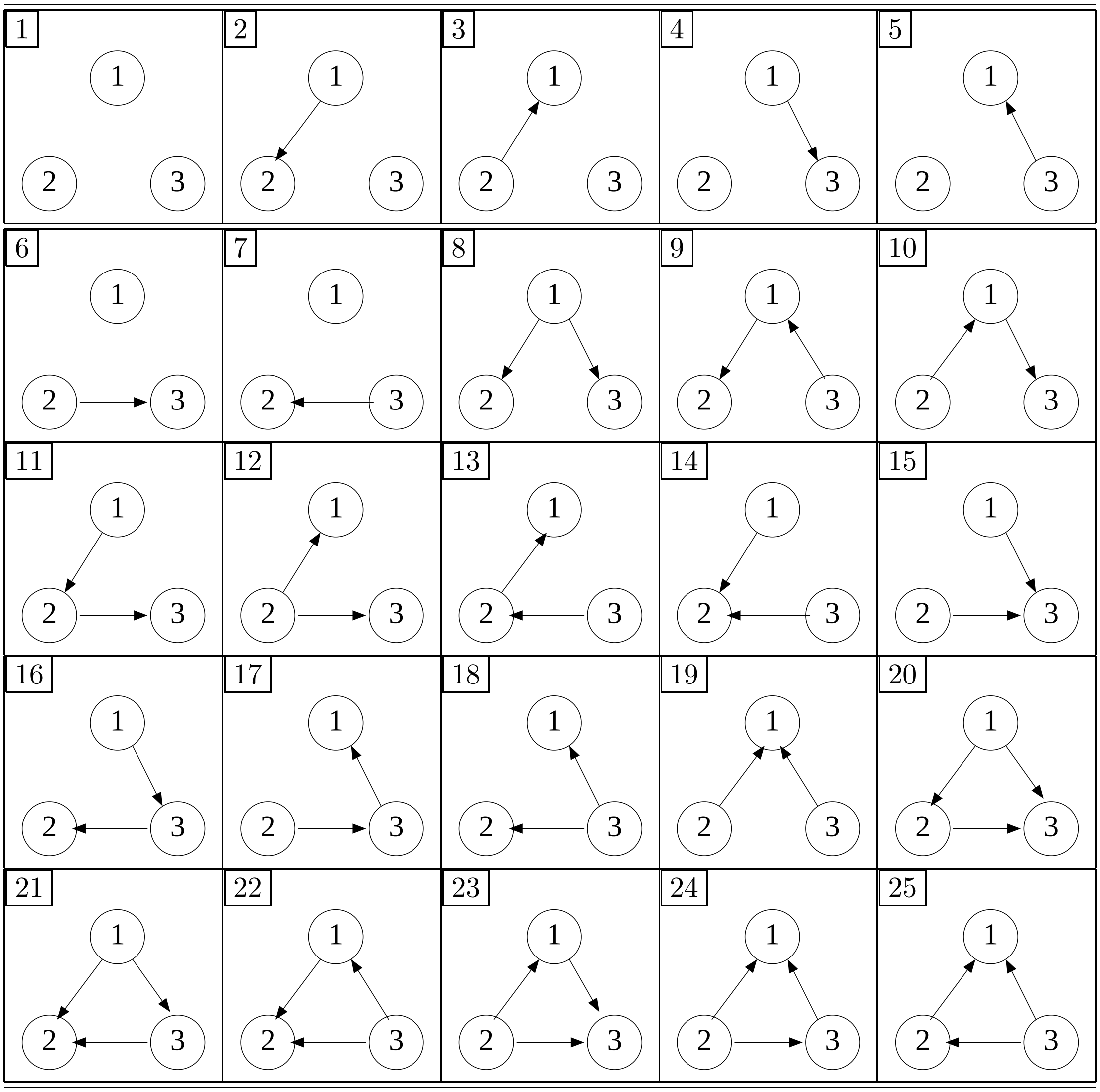
$$\begin{array}{rcl@{\quad}c@{\quad}rcl}
\theta_{1,1} & = & P(X_1 = 1), & \cdots & \theta_{1,5} & = & P(X_1 = 5) \\
\theta_{2,1,1} & = & P(X_2 = 1|X_1 = 1), & \cdots & \theta_{2,1,5} & = & P(X_2 = 5|X_1 = 1) \\
\theta_{2,2,1} & = & P(X_2 = 1|X_1 = 2), & \cdots & \theta_{2,2,5} & = & P(X_2 = 5|X_1 = 2) \\
\theta_{2,3,1} & = & P(X_2 = 1|X_1 = 3), & \cdots & \theta_{2,3,5} & = & P(X_2 = 5|X_1 = 3) \\
\theta_{3,1,1} & = & P(X_3 = 1|X_1 = 1), & \cdots & \theta_{3,1,5} & = & P(X_3 = 5|X_1 = 1) \\
\theta_{3,2,1} & = & P(X_3 = 1|X_1 = 2), & \cdots & \theta_{3,2,5} & = & P(X_3 = 5|X_1 = 2) \\
\theta_{3,3,1} & = & P(X_3 = 1|X_1 = 3), & \cdots & \theta_{3,3,5} & = & P(X_3 = 5|X_1 = 3) \\
 & \vdots & & \cdots & & \vdots &
\end{array}$$

(Note that $\theta_{1,6} = 1 - \sum_{i=1}^{5} \theta_{1,i}$ and $\theta_{i,j,6} = 1 - \sum_{k=1}^{5} \theta_{i,j,k}$ for $i \in \{2, 3, 4, 5, 6\}$ and $j \in \{1, 2, 3, 4, 5, 6\}$.)

For convenience, values for these probabilities were simulated from Dirichlet distributions with uniform hyperparameters on the interval $[0, 5]$.

We simulated (X_1, X_2, X_3) by assigning values to X_1 and then assigning values to X_2 and

Table 2.2: Directed Acyclic Graphs on Three Nodes



X_3 given X_1 according to the probabilities above. For this specific example we generated 100,000 values. Given this list of data, we used the AIC and BIC scoring mechanisms to select network 8 out of the 25 possible DAGs in Table 2.2. As mentioned in Chapter 1, AIC and BIC will not be able to distinguish between networks that are in the same Markov equivalent class, so if a data set scored by AIC and BIC recovers a network in the same Markov class it is considered successful. The Markov classes for the 25 DAGs listed in Table 2.2 are: {1}, {2,3}, {4,5}, {6,7}, {8,9,10}, {11,12,13}, {14}, {15}), {16,17,18}, {19}, {20,21,22,23,24,25}. Table 2.3 shows the AIC and BIC scores for all 25 DAGs and a successful recovery of DAG 8.

Table 2.3: AIC and BIC Recovery

Graph Number (n)	AIC	BIC
1	1040886.085	1041057.317
2	1001501.922	1001958.543
3	1001501.961	1001958.543
4	991255.256	991711.879
5	991255.259	991711.879
6	1036182.434	1036639.054
7	1036182.434	1036639.054
8	951871.096	952613.104
9	951871.096	952613.104
10	951871.096	952613.104
11	996798.271	997540.279
12	996798.271	997540.279
13	996798.271	997540.279
14	991435.745	936604.692
15	986551.608	987293.616
16	986551.608	987293.616
17	986551.608	987293.616
18	956755.233	958924.180
19	952051.582	954505.917
20	952051.582	954505.917
21	952051.582	954505.917
22	952051.582	954505.917
23	952051.582	954505.917
24	952051.582	954505.917
25	952051.582	954505.917

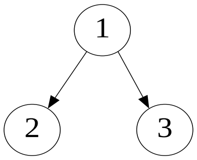
2.5 Discretizing Data

The objective of discretizing data in the context of Bayesian networks is to force continuous data into the multinominal model. This process involves reassigning all values in a particular interval to a single value. To illustrate the discretizing data process, we spend a significant portion of this thesis discussing the aggregation of values in an already discrete data set to a smaller number of values. We will still refer to this “discretization of discrete data.” as discretization.

2.5.1 Discretizing Discrete Data

To illustrate the concern in discretizing already discrete data consider the DAG in Figure 2.1.

Figure 2.1: A Three Node DAG



Now consider the discretization defined for $i = 1, 2, 3$, let

$$Y_i = \begin{cases} 1 & , \text{ if } X_i \in \{1,2\} \\ \\ 2 & , \text{ if } X_i = 3. \end{cases} \tag{2.5}$$

Our goal is to now show that the conditional independence of nodes 2 and 3 given node 1 is not preserved. From above we have:

$$\begin{aligned} P(Y_2 = 1, Y_3 = 1 | Y_1 = 1) &= P(X_2 \in \{1,2\}, X_3 \in \{1,2\} | X_1 \in \{1,2\}) \\ &= \tfrac{P(X_1 \in \{1,2\}, X_2 \in \{1,2\}, X_3 \in \{1,2\})}{P(X_1 \in \{1,2\})} \\ &= \tfrac{P(X_2 \in \{1,2\}, X_3 \in \{1,2\} | X_1 = 1) P(X_1 = 1) + P(X_2 \in \{1,2\}, X_3 \in \{1,2\} | X_1 = 2) P(X_1 = 2)}{P(X_1 \in \{1,2\})}. \end{aligned}$$

Since X_2 and X_3 are conditionally independent given X_1, we can factor the numerator to get

$$\begin{aligned} P(Y_2 = 1, Y_3 = 1 | Y_1 = 1) &= \tfrac{P(X_2 \in \{1,2\} | X_1 = 1) P(X_3 \in \{1,2\} | X_1 = 1) P(X_1 = 1)}{P(X_1 \in \{1,2\})} \\ &\quad + \tfrac{P(X_2 \in \{1,2\} | X_1 = 2) P(X_3 \in \{1,2\} | X_1 = 2) P(X_1 = 2)}{P(X_1 \in \{1,2\})} \end{aligned}$$

Similarly one can show that:

$$P(Y_2 = 1 | Y_1 = 1) = \frac{P(X_2 \in \{1,2\} | X_1 = 1) P(X_1 = 1) + P(X_2 \in \{1,2\} | X_1 = 2) P(X_1 = 2)}{P(X_1 = 1) + P(X_1 = 2)}$$

and

$$P(Y_3 = 1|Y_1 = 1) = \frac{P(X_3 \in \{1,2\}|X_1 = 1)P(X_1 = 1) + P(X_3 \in \{1,2\}|X_1 = 2)P(X_1 = 2)}{P(X_1 = 1) + P(X_1 = 2)}$$

Combining the equalities above we see that:

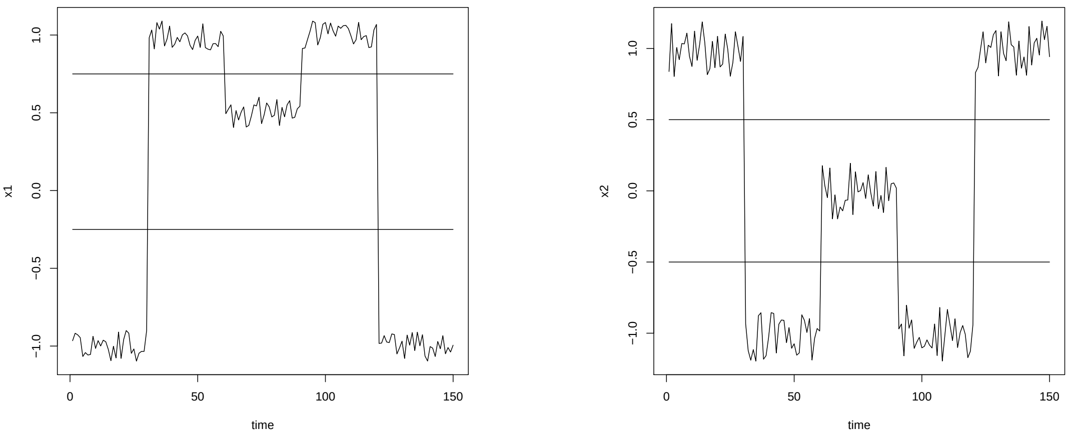
$$P(Y_2 = 1, Y_3 = 1|Y_1 = 1) \neq P(Y_2 = 1|Y_1 = 1) \cdot P(Y_3 = 1|Y_1 = 1) \tag{2.6}$$

Thus, nodes 2 and 3 are not conditionally independent given node 1.

2.5.2 Discretizing Continuous Data

An example of ad hoc discretization of continuous data can be found in [19]. In this thesis, the authors had simulated time series data for several nodes similar to that depicted in Figure 2.2. There was a fairly obvious relationship between the random variables X_1 and X_2 in a "high/low sense", but once high or low, each random variable was then, independently, augmented by Gaussian noise.

Figure 2.2: An Example of Loss of Conditional Independence



The discretizations used divided up the data (the y-axis) into the dominant regimes of low, middle, and high range values (assigned as 1's, 2's, and 3's) as depicted by the breaks determined by the horizontal lines in Figure 2.2. However, the independence of the noise components were completely lost and the resulting data appeared to have completely deterministic relationships. For example, every instance of X_1 as 1 was matched by an instance of X_2 as -1. This sort of overly coarse discretization, resulting in a loss of information about dependence or conditional independence, appears typical among practitioners in the literature.

Chapter 3

MINIMUM DESCRIPTION LENGTH FOR BAYESIAN NETWORK

3.1 Introduction

Arguably, the point of statistical modeling is to find regularities in an observed data set. Discovered regularities allow the modeler to be able to describe the data more succinctly. The minimum description length (MDL) principle, introduced in 1978 by Jorma Rissanen [23], is a model selection technique that chooses, as the best model, the one that permits the shortest encoding of both the model and the observed data.

In this section, we describe how Friedman and Goldszmidt [11] define a description length that can be used for network recovery from discrete data. It is the approximate length of storage space, measured in bits, for the binary representation of the DAG (network structure), the parameters (the θ_{ijk}) that, together with the DAG, define a Bayesian network, and the data itself. We will see, in the end, that it is simply another penalized likelihood approach.

In what follows, we repeatedly use the fact that an integer k can be encoded in approximately $\lceil \log_2 k \rceil$ bits. For example, the decimal value 9 becomes, in binary, the $\lceil \log_2 9 \rceil = 4$ bit number 1001. Throughout this thesis, we will use log to denote the base 2 logarithm, though, in the end the base is unimportant for the comparisons we will make.

3.2 Description Length for Bayesian Networks

Friedman and Goldszmidt [11] suggest an approach based on the **minimum description length principle** that chooses a discretization which preserves the information in the original data set. Description length, or the “minimum description length (MDL) score”, for a Bayesian network is the number of bits required to **encode the network,** including (1) the DAG, (2) the variables, and (3) the parameters, and to **encode the data**. In this Section, we follow the construction of Friedman and Goldszmidt [11] to derive the MDL score.

3.2.1 Encoding the Network

Encoding Variables

We need to store the number of variables and the number of possible values taken on by each variable. We assume that X_i, can take on $||X_i||$ possible values, and that they are integers ranging from 1 to $||X_i||$. (For example, if X_1 can take on values in $\{2, 5, 11\}$, we would relabel the values as 1, 2, and 3.)

The approximate number of bits needed to encode the number of nodes/variables n, and the number of possible values taken on by each of those variables is

$$\log n + \sum_{i=1}^{n} \log ||X_i||. \tag{3.1}$$

Encoding DAG

In order to completely describe the network structure, we must also include the number of parents for each node and the actual list of parents for each node. As a simplification, since the number of parents of node i, which is denote by $|\Pi_i|$, is always less than n, we can, conservatively, reserve $\log n$ bits to encode $|\Pi_i|$. Doing this for each node, we add

$$\sum_{i=1}^{n} \log n = n \log n \tag{3.2}$$

to our network description length.

For the actual list of parents, since the maximum value in the list of indices is n, we will use the conservative value of $\log n$ bits to encode each index. For node i, we must encode $|\Pi_i|$ different indices, each using a length of $\log n$ bits of space. So, in total, to encode all parent lists, we will use

$$\sum_{i=1}^{n} |\Pi_i| \cdot \log n \tag{3.3}$$

bits.

In total, we use

$$\log n + \sum_{i=1}^{n} \log ||X_i|| + \sum_{i=1}^{n} (1 + |\Pi_i|) \cdot \log n \tag{3.4}$$

bits to encode the DAG structure.

Encoding Parameters

The Bayesian network consists of a DAG together with a collection of parameters $\theta_{ijk} = P(X_i = k|\Pi_i = j)$. Since $\sum_k \theta_{ijk} = 1$, we only need to encode $||\Pi_i|| \cdot (||X_i|| - 1)$ parameters for node i. However, the parameters are not integer valued. In our case, for network recovery, we will actually be storing/encoding parameters that have been estimated from our m n-dimensional data points. Friedman and Goldszmidt [11] indicate that the "usual choice in the literature" is to use $\frac{1}{2}\log m$ bits per parameter. Thus, we will use

$$\frac{1}{2} \log m \cdot \sum_{i=1}^{n} ||\Pi_i|| \cdot (||X_i|| - 1) \tag{3.5}$$

bits to encode the estimated network parameters.

In summary, the total description length (and contribution to the MDL score) **for the network** is

$$\begin{aligned} \widetilde{DL}_{net} &= \log n + \textstyle\sum_{i=1}^{n} \log ||X_i|| + \sum_{i=1}^{n} (1 + |\Pi i|) \log n + \frac{1}{2} \log m \sum_{i=1}^{n} ||\Pi_i|| (||X_i|| - 1) \\ &= \log n + \textstyle\sum_{i=1}^{n} (\log ||X_i|| + (1 + |\Pi_i|) \log n) + \frac{1}{2} \log m \sum_{i=1}^{n} ||\Pi_i|| (||X_i|| - 1). \end{aligned}$$

Since we are trying to infer the best connecting arrows for a graph on n node, we presumably already know the number of nodes and would not be encoding it. Thus, we drop the $\lceil \log n \rceil$ term and define the **network description length as**

$$DL_{net} = \sum_{i=1}^{n} \log ||X_i|| + \sum_{i=1}^{n} (1 + |\Pi_i|) \cdot \log n + \frac{1}{2} \log m \cdot \sum_{i=1}^{n} ||\Pi_i|| \cdot (||X_i|| - 1). \tag{3.6}$$

Note that this is not explicitly dependent on the θ parameters.

3.2.2 Encoding Data

Consider the string of digits 2213 stored, in binary as 1010111. (The binary representations of 1, 2, and 3 are 1, 10, and 11, respectively.) Without separators, the binary string 1010111 can not be decoded near the end. The 111 maybe be, in decimal, three 1's, a 1 followed by a 3, or a 3 followed by a 1. "Prefix codes" avoid this problem by encoding the digits with zeros and ones in a way so that no encoded digit is a prefix of any other encoded digit. Further consideration can be made to ensure the code is not only a prefix code, but one that results in the maximum compression of the data by assigning, to the original decimal digits, binary codes of lengths inversely proportional to the frequencies with which they appear in the original data.

In the context of Bayesian networks, we wish to encode the values in the $(m \times n)$-dimensional data set with code lengths that are inversely proportional to the frequencies (equivalently, estimated probabilities) with which they appear in the data. Friedman and Goldszmidt [11] use **Shannon coding** [6] which is not optimal in terms of compression, but which encodes each n-dimensional data point $\vec{x} = (x_1, x_2, \ldots, x_n)$ using approximately $-\log p(\vec{x})$ bits. Thus, the entire data set, consisting of m such vectors is encoded in approximately

$$DL_{data} = -\sum_{i=1}^{m} \log p(\vec{x}_i) \tag{3.7}$$

bits. This is desirable from a modeling standpoint since it corresponds to the familiar log-likelihood commonly used in statistical inference.

Summing (3.6) and (3.7), we define the description length for a Bayesian network as

$$MDL = DL_{net} + DL_{data}.$$

Or,

$$DL = \sum_{i=1}^{n} \log ||X_i|| + \sum_{i=1}^{n} (1 + |\Pi_i|) \cdot \log n + \frac{1}{2} \log m \cdot \sum_{i=1}^{n} ||\Pi_i|| \cdot (||X_i|| - 1) - \sum_{i=1}^{m} \log p(\vec{x}_i). \tag{3.8}$$

This score is similar to the AIC and BIC scores in that it is also a negative log-likelihood plus a term (DL_{net}) that penalizes for the number of parameters.

Given discrete data and a collection of possible Bayesian networks, the network chosen by the MDL principle is the one that minimizes (3.8). In light of the form of DL_{data}, we see that this is simply a penalized log-likelihood scoring metric, similar to the AIC and BIC discussed in Chapter 2. We now illustrate the performance of all three for a three node network. The list of all 25 DAGs corresponding to 3 node networks can be found in Table 2.2 and we will refer to these DAGs as they are numbered here. We fixed parameters θ_{ijk} for DAG 8 in Table 2.2 shown in Chapter 2 and simulated $100,000$ values of (X_1, X_2, X_3). (Each X_i was assumed to take on values in $\{1, 2, \ldots, 6\}$.)

Results are shown in Table 3.1 along with the previously computed values of AIC and BIC. The AIC, BIC, and MDL scores all recovered the correct network "up to Markov equivalence". Recovering the specific graph within a Markov equivalence class requires experimental as opposed to simply observed data and is not the subject of this paper.

3.3 Minimum Description Length for Discretization

A discretization of data for the random variable X_i, represented by node i, is a mapping from the range of values in the data set to the set $\{1, 2, \ldots, k_i\}$ for some $k_i \geq 1$. It can be described by ordering the distinct values observed for X_i and inserting up to $k_i - 1$ "thresholds". For example

$$\underbrace{0.38 \quad 0.42 \quad 0.53 \quad 0.71}_{\text{map to 1}} \;|\; \underbrace{1.37 \quad 1.94 \quad 2.10}_{\text{map to 2}} \;|\; \underbrace{5.38 \quad 7.11}_{\text{map to 3}}. \tag{3.9}$$

For simplicity, we will refer to node i and the random variable X_i interchangeably. Also, for simplicity, we will assume at this point that node i is the only continuous one in the network and that the others are discrete or have already gone through a discretization process. Furthermore, we assume that X_i takes on m_i distinct values in the data set with m n-dimensional points. (Clearly $m_i \leq m$ with equality in the case of truly continuous data.) The discretized version of X_i will be denoted by X_i^* and we will use k_i to denote the number of values taken on by X_i^*.

Friedman and Goldszmidt augment the MDL score with description lengths for

- the discretization rule which consists of thresholds for mapping data to $\{1, 2, \ldots, k\}$,

Table 3.1: AIC, BIC and MDL Recovery

Graph Number (n)	AIC	BIC	MDL
1	1040886.085	1041057.317	520443.714
2	1001501.922	1001958.543	500885.425
3	1001501.961	1001958.543	500885.425
4	991255.256	991711.879	495762.093
5	991255.259	991711.879	495762.093
6	1036182.434	1036639.054	518225.681
7	1036182.434	1036639.054	518225.681
8	951871.096	952613.104	476213.804
9	951871.096	952613.104	476213.804
10	951871.096	952613.104	476213.804
11	996798.271	997540.279	498677.392
12	996798.271	997540.279	498677.392
13	996798.271	997540.279	498677.392
14	991435.745	936604.692	496536.904
15	986551.608	987293.616	493554.060
16	986551.608	987293.616	493554.060
17	986551.608	987293.616	493554.060
18	956755.233	958924.180	479196.648
19	952051.582	954505.917	476988.615
20	952051.582	954505.917	476988.615
21	952051.582	954505.917	476988.615
22	952051.582	954505.917	476988.615
23	952051.582	954505.917	476988.615
24	952051.582	954505.917	476988.615
25	952051.582	954505.917	476988.615

- the description of the discretized data, and
- the description of the original data set based on the discretized data set.

For a fixed DAG and a fixed threshold assignment, we proceed as follows.

As before, we assume that we have m observations of the n-dimensional $(X_1, X_2, \ldots, X_n)$. Let m_i be the number of distinct values taken on by X_i in the data set. Note that $m_i \leq m$, with equality possible only for truly continuous data. Define X_i^* as the discretized version of X_i. To discretize to $k_i \leq m_i$ values, we need to choose $k_i - 1$ thresholds to put in $m_i - 1$ spaces between ordered values.

There are $\begin{pmatrix} m_i - 1 \\ k_i - 1 \end{pmatrix}$ threshold configurations to consider. Since we will not know in advance how many values we should have in the discretized data set, we need to consider everything from $k_i = 1$, which corresponds to mapping all values for X_i in the data set to the single value of 1, to $k_i = m_i$, which corresponds to no discretization at all. In total, there are

$$\begin{pmatrix} m_i - 1 \\ 0 \end{pmatrix} + \begin{pmatrix} m_i - 1 \\ 1 \end{pmatrix} + \ldots \begin{pmatrix} m_i - 1 \\ m_i - 1 \end{pmatrix} = 2^{m_i - 1}$$

discretizations to consider.

Friedman and Goldszmidt [11] define a description length score for a network with X_i discretized into a particular configuration of k_i thresholds using essentially four terms. These terms include DL_{net} and DL_{data}, previously described in (3.6) and (3.7), computed now after discretization– we will call these terms DL^*_{net} and DL^*_{data}. Also included are terms that encode the index denoting a particular discretization policy and description length for information needed to recover the original data from the discretized data.

3.3.1 Encoding the Discretization Policy

For fixed m_i and k_i, there are $\begin{pmatrix} m_i - 1 \\ k_i - 1 \end{pmatrix}$ different possible configurations for thresholds. Assume we have labeled them from 1 to $\begin{pmatrix} m_i - 1 \\ k_i - 1 \end{pmatrix}$. Storing the index for a particular policy will take at most $\left\lceil \log \begin{pmatrix} m_i - 1 \\ k_i - 1 \end{pmatrix} \right\rceil$ bits. Friedman and Goldszmidt use a conservative upper bound based on the inequality

$$\begin{pmatrix} n \\ k \end{pmatrix} \le 2^{nH(k/n)},$$

where

$$H(p) := -p \log p - (1-p) \log(1-p),$$

and conservatively reserve

$$DL_{DP} = (m_i - 1)H\left(\frac{k_i - 1}{m_i - 1}\right) \tag{3.10}$$

to encode the discretization policy. (In practice, we define $H(0) = H(1) = 0$.)

3.3.2 Encoding Recovery of Original Data

Consider again the example of 9 ($m_i = 9$) distinct values for X_i given by (3.9) from a data set with $m \geq m_i$ values. Every time we assign a discretized value to an original value, we should store the original value for recovery. Instead, however, we will store the Shannon binary code for the original value using estimated conditional probabilities based on the entire data set. For example, the value 2.10 might appear in the entire data set one-fourth of the time. That is, $\widehat{P}(X_i = 2.10) = 1/4$. Among the instances of 1.37, 1.94, and 2.10, it might appear half of the time. That is, $\widehat{P}(X_i = 2.10|X_i^* = 2)$. Given a discretized value of 2, we will encode the original value using approximately $-\log \widehat{P}(X_i|X_i^* = 2)$ bits. In total, for recovering original data from discretized data, we add

$$DL_{rec} = -\sum_{i=1}^{m} \log \widehat{P}(X_i|X_i^*) \tag{3.11}$$

to the description length.

In summary, Friedman and Goldszmidt [11] define the **description length discretization score** as

$$DL^* = (m_i - 1)H\left(\frac{k_i - 1}{m_i - 1}\right) + DL_{net}^* + DL_{data}^* + DL_{rec}. \tag{3.12}$$

Given a data set and a particular network structure, one scores various discretization mappings for X_i and chooses the discretization that minimizes (3.12). For a given network, the score in (3.12) will change over discretizations only in terms directly linked to the ith node. Thus, as Friedman and Goldszmidt point out, we only need to consider the “local description length score” defined, by picking out relevant terms, as

$$\begin{aligned} DL_{local} &= (m_i - 1)H\left(\tfrac{k_i-1}{m_i-1}\right) + \log k_i \\ &\quad +\tfrac{1}{2}\log m\left[||\Pi_i||(k_i - 1) + \textstyle\sum_{j:X_i\in\Pi_j} ||\Pi_j^*||(||X_j|| - 1)\right] \\ &\quad -m\left[\widehat{I}(X_i^*, \Pi_i) + \textstyle\sum_{j:X_i\in\Pi_j} \widehat{I}(X_j, \Pi_j^*)\right]. \end{aligned} \tag{3.13}$$

Here, Π_j^* is a the set of parents for node j, denoted with an asterisk since it includes the discretized X_i^*, and,

$$\widehat{I}(\vec{X}, \vec{Y}) = \sum_{\vec{x},\vec{y}} \widehat{P}(\vec{X} = \vec{x}, \vec{Y} = \vec{y}) \cdot \log\left(\frac{\widehat{P}(\vec{X} = \vec{x}, \vec{Y} = \vec{y})}{\widehat{P}(\vec{X} = \vec{x})\widehat{P}(\vec{Y} = \vec{y})}\right) \tag{3.14}$$

is the estimated mutual information between random vectors $\vec{X}$ and $\vec{Y}$.

Chapter 4

A DISCRETIZATION SEARCH STRATEGY FOR BAYESIAN NETWORKS

4.1 Introduction

Friedman and Goldszmidt [11] have developed a method for discretization of continuous data for Bayesian networks which is based on the minimum description length principle. However, Computation of a single value of (3.13), which is based on network structure through the information terms, can be quite time consuming. For even moderately sized m_i, computation of (3.13) repeatedly to check all 2^{m_i-1} discretization policies can be prohibitive, and when multiple/all nodes need to be discretized, computation of (3.13) becomes almost impossible. Friedman and Goldszmidt [11] give some further computational simplifications and suggest a greedy search routine.

In this Chapter, we explore a structure of the minimum description length developed by Friedman and Goldszmidt [11] and then provide an alternative efficient search strategy for the smallest DL_{local} score. We assert that, for a single node discretization, one need only check m_i values of DL_{local} as opposed to 2^{m_i-1}. Multiple nodes can then be cycled for discretization just as Friedman and Goldszmidt have suggested.

4.2 Searching for Discretizations

A discretization of X_i involves putting thresholds between values in the data set. In the case of $m_i \leq m$ distinct values, there are

$$\begin{pmatrix} m_i - 1 \\ 0 \end{pmatrix} + \begin{pmatrix} m_i - 1 \\ 1 \end{pmatrix} + \ldots \begin{pmatrix} m_i - 1 \\ m_i - 1 \end{pmatrix} = 2^{m_i - 1}$$

different discretizations to consider. Clearly, this number can get quite large for large data sets with truly continuous ($m_i = m$) data points. We now illustrate the enumeration of discretizations in a toy case of categorization ("discretizing discrete data") where they can be explicitly listed. We choose a node with 6 distinct values ($m_i = 6$). Based on Section 3.3 in Chapter 3, we know that there are

$$\begin{pmatrix} m_i - 1 \\ k_i - 1 \end{pmatrix}$$

discretization policies that result in k_i categories. The total number of possible discretizations is $2^5 = 32$.

For notational simplicity, we assume, for the remainder of this paper, that the node to be discretized is labeled as node 1. Also, as we are comparing values of DL_{local} for various discretizations of X_1, we will drop all asterisk superscript notation, as it is understood that we are considering discretized values.

4.2.1 Single Threshold Top-Down Search Strategy

Let $DL_{local}(0)$ be (3.13) with all $m_1 - 1$ thresholds in place. For $j = 1, 2, \ldots, m_1 - 1$, let $DL_{local}(-j)$ be (3.13) with all thresholds except for the jth threshold.

In order to minimize DL_{local} over all $2^{m_1 - 1}$ discretization policies for X_1, make comparisons of $DL_{local}(0)$ with $DL_{local}(-j)$ for $j = 1, 2, \ldots, m_1 - 1$. If $DL_{local}(-j) \leq DL_{local}(0)$, remove the jth threshold.

We first consider the effectiveness of this search strategy in the very ideal situation where we augment given discrete data by introducing superfluous values for node 1 in a larger discrete set. For example, we might replace values of 5 in the original data set with values in $\{5, 6, 7\}$ with some arbitrary probabilities, whereupon we hope that our search strategy will minimize DL_{local} and that the configuration of thresholds that does such will correctly map values in $\{5, 6, 7\}$ back to the original value of 5. We will prove, in this case, that the "single threshold top-down" search strategy will find the discretization policy that minimizes DL_{local} among all 2^{m_1-1} discretizations for a large enough sample size m.

Indeed, m, m_1, $||\Pi_1||$, and $||X_j||$ are constant, and $0 \leq H(p) \leq 1$. Note that

$$(m_1-1)H\left(\frac{k_1-1}{m_1-1}\right)+\log k_1+\frac{1}{2}\log m\left[||\Pi_1||(k_1-1)+\sum_{j:X_1\in\Pi_j}||\Pi_j||(||X_j||-1)\right] \tag{4.1}$$

will be constant over any discretization policy for X_1 with a fixed number of thresholds. In particular, when comparing all possible single threshold removals, starting with any fixed number of thresholds, we can restrict our attention to maximizing

$$\widehat{I}(X_1,\Pi_1)+\sum_{j:X_1\in\Pi_j}\widehat{I}(X_j,\Pi_j). \tag{4.2}$$

In the next Section, we consider what our search strategy does to (4.2) in the absence of (4.1) in an ideal situation where there is a "correct" discretization. We will see that it maximizes (4.2) and that it does so while leaving a minimal number of thresholds. In Section 4.2.4, we will consider (4.1) and conclude that we are indeed minimizing DL_{local}. We will also see that it finds the correct discretization.

4.2.2 A Closer look at Information Terms in an Ideal Situation

As the notation to follow gets a bit cumbersome, we illustrate most claims in this Section and the next with a concrete example. Consider a two-node network where node 1 is a parent to node 2, and assume that that both nodes take on values in $\{1,2,3\}$. From m data points, we can produce estimates

$$\begin{aligned}\widehat{p}(i,j) &:= \widehat{P}(X_1=i,X_2=j)\\ \widehat{p}_1(i) &:= \widehat{P}(X_1=i),\text{ and}\\ \widehat{p}_2(j) &:= \widehat{P}(X_2=j),\end{aligned}$$

for $i,j,\in\{1,2,3\}$.

In general, for this two node network, we would assume that node 1 takes on values in $\{1,2,\ldots,m_1\}$ and node 2 takes on values in $\{1,2,\ldots,m_2\}$, and would work with the estimates $\widehat{p}(i,j)$, $\widehat{p}_1(i)$, and $\widehat{p}_2(j)$ for $i\in\{1,2,\ldots,m_1\}$ and $j\in\{1,2,\ldots,m_2\}$.

We now “explode” the data for our specific example at node 1 into values in $\{1,2,3,4,5,6\}$ by replacing instances of 1 with values in $\{1,2\}$ with probabilities 1/3 and 2/3, respectively, replacing original instances of 2 with values in $\{3,4,5\}$ with probabilities 2/7, 4/7, and 1/7, respectively, and replacing original instances of 3 with the value 6. Thus, in our “exploded data set”, node 1 is taking on values in $\{1,2,3,4,5,6\}$ with probabilities denoted as $\widetilde{p}_1(i)$ for $i = 1,2,3,4,5,6$, where, for example, $\widetilde{p}_1(1) = \frac{1}{3}\widehat{p}_1(1)$ and $\widetilde{p}_1(4) = \frac{4}{7}\widehat{p}_1(2)$. Joint probabilities for X_1 and X_2 are denoted by $\widetilde{p}(i,j)$ where we have, for example,

$$\widetilde{p}(1,j) = \frac{1}{3}\widehat{p}(1,j) \qquad \text{and} \qquad \widetilde{p}(4,j) = \frac{4}{7}\widehat{p}(2,j).$$

In the more general case, we could “explode” the data at node 1 more generally by replacing instances of 1 with values in $\{1,2,\ldots,\ell_1\}$ some probabilities $q(1,1), q(1,2),\ldots,q(1,\ell_1)$, summing to 1, replacing original instances of 2 with values in $\{\ell_1+1,\ell_1+2,\ldots,\ell_1+\ell_2\}$ with respective probabilities $q(2,1), q(2,2),\ldots,q(2,\ell_2)$, summing to 1, and so forth.

By “exploding” the data at node one, we have introduced superfluous values for X_1 in terms of probabilities for X_2. For example, $\widetilde{P}(X_2 = j|X_1 = 1) = \widetilde{P}(X_2 = j|X_1 = 2)$ for all $j \in \{1,2,3\}$. Any proper discretization process should aggregate the values 1 and 2 for X_1 back into one value. Here, $\widetilde{P}(X_1 = i, X_2 = j)$ is used to denote the probability $\widetilde{p}(i,j)$. Similarly, $\widetilde{p}_1(i)$ may be denoted as $\widetilde{P}(X_1 = i)$.

For this two-node network, (4.2) is simply $\widehat{I}(X_1, X_2)$, which we will denote by $\widehat{I}$. Define this estimated information as

$$\widehat{I} := \sum_{i,j} \widehat{I}_{i,j} \tag{4.3}$$

where

$$\widehat{I}_{ij} := \widehat{p}(i,j) \cdot \log\left(\frac{\widehat{p}(i,j)}{\widehat{p}_1(i) \cdot \widehat{p}_2(j)}\right) \tag{4.4}$$

and the sums run over $i \in \{1,2,3\}$ and $j \in \{1,2,3\}$.

Define the corresponding information $\widetilde{I}$ and information terms $\widetilde{I}_{i,j}$ using $\widetilde{p}(i,j)$ in place of $\widehat{p}(i,j)$ with sums running over $i \in \{1,2,\ldots,6\}$ and $j \in \{1,2,3\}$.

It is easy to verify information term relationships such as

$$\widetilde{I}_{1,j} = \frac{1}{3}\widehat{I}_{1,j} \qquad \text{and} \qquad \widetilde{I}_{4,j} = \frac{4}{7} \cdot \widehat{I}_{2,j},$$

and consequently that

$$\widetilde{I} = \sum_{\substack{i \in \{1,2,\ldots 6\} \\ j \in \{1,2,3\}}} \widetilde{I}_{ij} = \sum_{\substack{i \in \{1,2,3\} \\ j \in \{1,2,3\}}} \widehat{I}_{ij} = \widehat{I}.$$

That is, we did not change the information between X_1 and X_2 by exploding the data.

We now show that our single threshold top-down search strategy will correctly recover the original values for X_1. We call this the "correct discretization", denoted as 12|345|6, which means that we will remove three thresholds from the "full discretization", denoted as 1|2|3|4|5|6, and that the values 1 and 2 will map back to 1, the values in $\{3, 4, 5\}$ will map back to 2, and the value 6, will map back to 3. We will assume that the original values in $\{1, 2, 3\}$ are all distinct in the sense that $\widehat{P}(X_2 = j|X_1 = 1) \neq \widehat{P}(X_2 = j|X_1 = 2)$ for some j, $\widehat{P}(X_2 = j|X_1 = 1) \neq \widehat{P}(X_2 = j|X_1 = 3)$ for some j, and $\widehat{P}(X_2 = j|X_1 = 2) \neq \widehat{P}(X_2 = j|X_1 = 3)$ for some j, so that they should not be aggregated further.

4.2.3 Removing a Threshold: The Impact on Information

Starting with the exploded data, with values for X_1 represented as 1|2|3|4|5|6, we consider the (X_1, X_2) information term after removing the threshold between the values r and $r+1$ for some $r \in \{1, 2, \ldots, 5\}$. Note that removal of this rth threshold will leave all values below the threshold unchanged, while all values above will be decreased by 1. For example, if we remove the third threshold, we denote the new configuration as 1|2|34|5|6, but it represents a mapping

$$\underbrace{1}_{\text{map to 1}} \quad | \quad \underbrace{2}_{\text{map to 2}} \quad | \quad \underbrace{3\ 4}_{\text{map to 3}} \quad | \quad \underbrace{5}_{\text{map to 4}} \quad | \quad \underbrace{6}_{\text{map to 5}} \quad .$$

Define, for $i \in \{1, 2, \ldots, 5\}$ and $j \in \{1, 2, 3\}$, the joint and marginal probabilities for X_1 and X_2 after the rth threshold is removed as $p^{(r)}(i, j)$, $p_1^{(r)}(i)$, and $p_2^{(r)}(j)$. We have, for $j \in \{1, 2, 3\}$, the relationships

$$\begin{aligned}
p^{(r)}(i,j) &= \widetilde{p}(i,j), \qquad i \in \{1,2,\ldots,r-1\} \;\; (r>1) \\
p^{(r)}(r,j) &= \widetilde{p}(r,j) + \widetilde{p}(r+1,j) \\
p^{(r)}(i,j) &= \widetilde{p}(i+1,j), \qquad i \in \{r+1,r+2,\ldots,6\}, \;\; (r<5)
\end{aligned}$$

$$\begin{aligned}
p_1^{(r)}(i) &= \widetilde{p}_1(i), \qquad i \in \{1,2,\ldots,r-1\} \\
p_1^{(r)}(r) &= \widetilde{p}_1(r) + \widetilde{p}_1(r+1) \\
p_1^{(r)}(i) &= \widetilde{p}_1(i+1), \qquad i \in \{r+1,r+3,\ldots,6\}, \;\; (r<5),
\end{aligned}$$

and

$$p_2^{(r)}(j) = \widetilde{p}_2(j) = \widehat{p}_2(j).$$

Defining $I^{(r)}$ and $I_{ij}^{(r)}$ analogous to (4.3) and (4.4), using $p^{(r)}(i,j)$, we have, for $j \in \{1,2,3\}$,

$$\begin{aligned}
I_{rj}^{(r)} &= p^{(r)}(r,j) \cdot \log\left(\tfrac{p^{(r)}(r,j)}{p_1^{(r)}(r)p_2^{(r)}(j)}\right) \\
&= [\widetilde{p}(r,j) + \widetilde{p}(r+1,j)] \cdot \log\left(\tfrac{\widetilde{p}(r,j)+\widetilde{p}(r+1,j)}{[\widetilde{p}_1(r)+\widetilde{p}_1(r+1)]\cdot\widetilde{p}_2(j)}\right) \\
&\leq \widetilde{p}(r,j) \cdot \log\left(\tfrac{\widetilde{p}(r,j)}{\widetilde{p}_1(r)\cdot\widetilde{p}_2(j)}\right) + \widetilde{p}(r+1,j)] \cdot \log\left(\tfrac{\widetilde{p}(r+1,j)}{\widetilde{p}_1(r+1)\cdot\widetilde{p}_2(j)}\right) \\
&= \widetilde{I}_{rj} + \widetilde{I}_{rj}.
\end{aligned}$$

The inequality is is due to the log-sum inequality,

$$\sum_{i=1}^{n} a_i \log\left(\frac{a_i}{b_i}\right) \leq \left[\sum_i a_i\right] \log\left(\frac{\sum_i a_i}{\sum_i b_i}\right),$$

which holds for any nonnegative $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$. The log-sum inequality can be shown to be an equality if and only if the a_i/b_i are equal for all $i = 1, 2, \ldots, n$. Thus, we have that

$$I_{rj}^{(r)} \leq \widetilde{I}_{rj} + \widetilde{I}_{r+1,j} \tag{4.5}$$

for $j \in \{1,2,3\}$, with equality if and only if

$$\frac{\widetilde{p}(r,j)}{\widetilde{p}_1(r)\cdot\widetilde{p}_2(j)} = \frac{\widetilde{p}(r+1,j)}{\widetilde{p}_1(r+1)\cdot\widetilde{p}_2(j)}.$$

This happens if and only if

$$\widetilde{P}(X_2 = j|X_1 = r) = \widetilde{P}(X_2 = j|X_1 = r+1). \tag{4.6}$$

for $j \in \{1, 2, 3\}$, which is precisely when the values r and $r+1$ in the exploded version of X_1 should be aggregated or discretized into one value. If we do not have (4.6), aggregating the values will result in a loss of information.

Note that $I^{(r)}$ denotes the value of the information between X_1 and X_2 with the rth threshold in the explosion for X_1 removed. In our example,

$$\widetilde{P}(X_2 = j|X_1 = 1) = \frac{\widetilde{p}(1, j)}{\widetilde{p}_1(1)} = \frac{(1/3)\widehat{p}(1, j)}{(1/3)\widehat{p}_1(1)} = \widehat{P}(X_2 = j|X_1 = 1)$$

and

$$\widetilde{P}(X_2 = j|X_1 = 2) = \frac{\widetilde{p}(2, j)}{\widetilde{p}_2(1)} = \frac{(2/3)\widehat{p}(1, j)}{(2/3)\widehat{p}_1(1)} = \widehat{P}(X_2 = j|X_1 = 1).$$

Thus, we have (4.6) and consequently

$$\begin{aligned} I^{(1)} &= \textstyle\sum_{i=1}^{5}\sum_{j=1}^{3} I_{ij}^{(1)} \\ &= \textstyle\sum_{j=1}^{3} I_{1j}^{(1)} + \sum_{i=2}^{5}\sum_{j=1}^{3} I_{ij}^{(1)} \\ &= \textstyle\sum_{j=1}^{3} I_{1j}^{(1)} + \sum_{i=3}^{6}\sum_{j=1}^{3} \widetilde{I}_{ij} \\ &= \textstyle\sum_{j=1}^{3}(\widehat{I}_{1j} + \widetilde{I}_{2j}) + \sum_{i=3}^{6}\sum_{j=1}^{3} \widetilde{I}_{ij} \\ &= \textstyle\sum_{i=1}^{6} \widetilde{I}_{ij} = \widetilde{I}. \end{aligned}$$

On the other hand, considering the second threshold removal from the full discretization 1|2|3|4|5|6, similar calculations show that

$$\widetilde{P}(X_2 = j|X_1 = 2) = \widehat{P}(X_2 = j|X_1 = 1)$$

which is not, in general equal to

$$\widetilde{P}(X_2 = j|X_1 = 3) = \widehat{P}(X_2 = j|X_1 = 2),$$

so we have the strict inequality $I^{(2)}_{2,j} < \widetilde{I}_{2,j} + \widetilde{I}_{3j}$ and therefore

$$\begin{aligned}
I^{(2)} &= \textstyle\sum_{i=1}^{5}\sum_{j=1}^{3} I^{(2)}_{ij} \\
&= \textstyle\sum_{j=1}^{3} I^{(2)}_{1j} + \sum_{j=1}^{3} I^{(2)}_{2j} + \sum_{i=3}^{5}\sum_{j=1}^{3} I^{(2)}_{ij} \\
&= \textstyle\sum_{j=1}^{3} \widetilde{I}_{1j} + \sum_{j=1}^{3} I^{(2)}_{2j} + \sum_{i=4}^{6}\sum_{j=1}^{3} \widetilde{I}_{ij} \\
&< \textstyle\sum_{j=1}^{3} \widetilde{I}_{1j} + \sum_{j=1}^{3}(\widetilde{I}_{2j} + \widetilde{I}_{3j}) + \sum_{i=4}^{6}\sum_{j=1}^{3} \widetilde{I}_{ij} \\
&= \textstyle\sum_{i=1}^{6} \widetilde{I}_{ij} = \widetilde{I}.
\end{aligned}$$

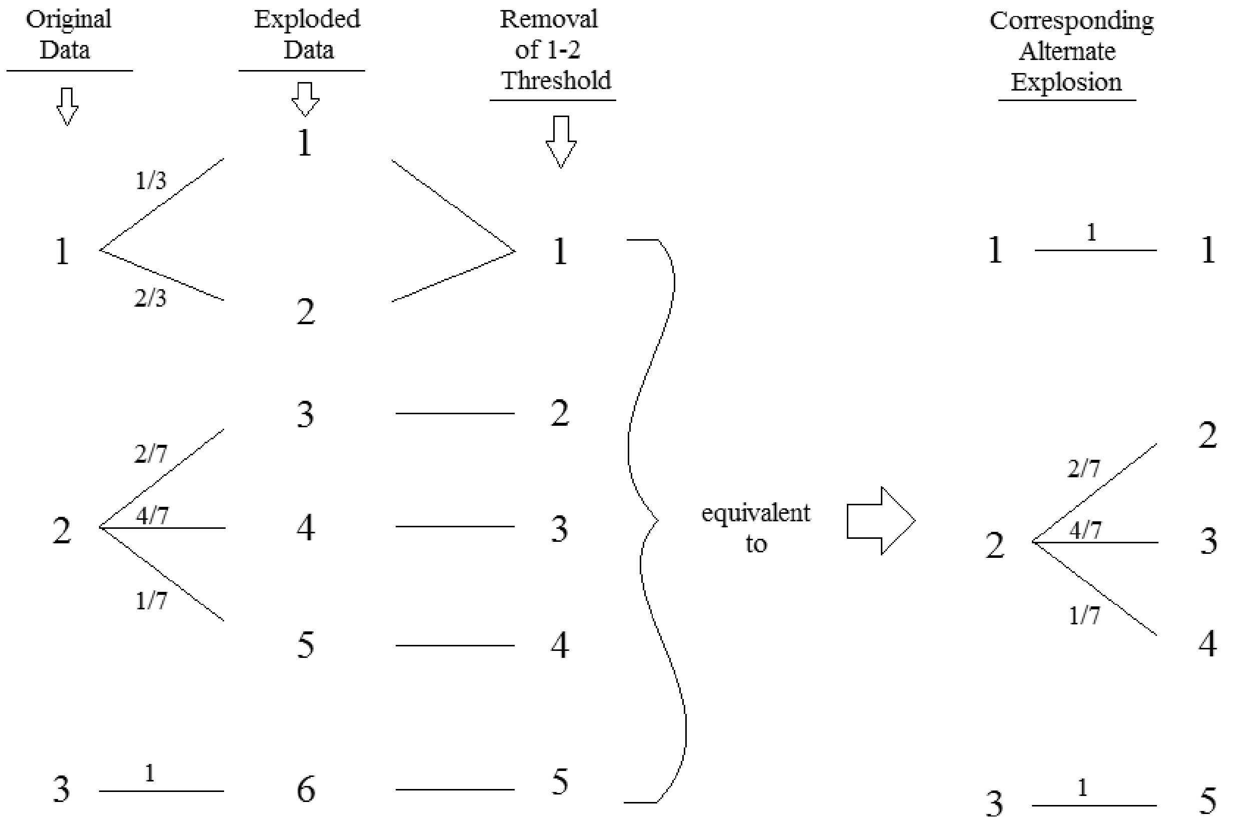
In all, we can show in this way that

$$I^{(1)} = \widetilde{I},\;\; I^{(2)} < \widetilde{I},\;\; I^{(3)} = \widetilde{I},\;\; I^{(4)} = \widetilde{I},\;\; \text{and}\;\; I^{(5)} < \widetilde{I}. \tag{4.7}$$

So, our search strategy will produce the correct discretization 12|345|6. The question remains though as to whether this is actually the discretization that minimizes DL_{local}. Due to (4.5), we will always have $I^{(r)} \leq \tilde{I}$. In fact, due to the log-sum inequality, any removal of a threshold from any configuration with any number of thresholds can never increase information. Thus, the full discretization 1|2|3|4|5|6 will always have maximal information. From (4.7), we see that the discretizations 12|3|4|5|6, 1|2|34|5|6, and 1|2|3|45|6 have the same, and thus also maximal, information. Note that each of these three configurations corresponds to a different explosion of the original data. This is illustrated for 12|3|4|5|6 in Figure 4.1.

Thus, by the same information arguments above, "correct removal" (removal of one of the superfluous thresholds) of a threshold from 12|3|4|5|6 will result in another configuration with the same maximal information. Since this is true starting with any one of the three correct single threshold removal configurations, we see that we can remove two of the superfluous thresholds from the full explosion and still maintain the maximal information. Continuing this argument, we can remove all surperfluous thresholds from the full discretization and the resulting configurations

Figure 4.1: Correct Removal of a Threshold Corresponds to an Alternate Explosion



of thresholds, which in this example is 12|345|6, will have the maximal information.

4.2.4 A Closer look at the Leading Terms in DL_{local}

For a general DAG, the leading terms in DL_{local}, given in (4.1) can be rewritten as

$$(m_1 - 1)H\left(\frac{k_1 - 1}{m_1 - 1}\right) + \log k_1 + \frac{1}{2}(\log m)(ck_1 - ||\Pi_1||) =: D(k_1)$$

for some $c \geq 0$ where $c = 0$ if and only if node 1 is not connected to any other nodes.

The second and third terms here are clearly increasing in k_1. The first term, as a function of k_1, is symmetric about $k_1 = (m_1 + 1)/2$, increasing to the left of this value and decreasing to the right. However, since

$$\frac{d}{dx}D(x)\bigg|_{x=k_1} = \log\left(\frac{m_1 - k_1}{k_1 - 1}\right) + \frac{1}{k_1} + \frac{1}{2}c \cdot \log m,$$

we can, when $c > 0$ choose m (the sample size) large enough to ensure that $D(k_1)$ is increasing in k_1. In this case, this first part of DL_{local} acts as a penalty term and DL_{local} is minimized by choosing the discretization that maximizes the information terms with the minimum number of thresholds. By design, the single threshold top-down search strategy will do exactly this.

4.2.5 More Complicated Networks

The observations in Section 4.2.4 were not dependent on the specific network, however, our analysis of the information terms was for a two-node network where X_1 was a parent to X_2. Since $\widehat{I}(X_1, X_2) = \widehat{I}(X_2, X_1)$, our search strategy will also find the "optimal discretization" for X_1, i.e. the one that that maximizes (4.2) with a minimum number of thresholds, for the two-node network where X_1 is a child of X_2.

We will now check that the strategy will find the optimal discretization for X_1 for general networks. To this end, we begin by independently considering the two types of terms in (4.2).

- $\widehat{I}(X_1, \Pi_1)$

Since replacing X_2 with a vector of random variables has no effect on any of the computations in Section 4.2.2, we see that our search strategy will find the discretization for X_1 that maximizes $\widehat{I}(X_1, \Pi_1)$ with a minimum number of thresholds.

- $\widehat{I}(X_j, \Pi_j)$ where $X_1 \in \Pi_j$

Suppose, for ease of exposition, that $j = 2$ and, that all nodes originally take values in $\{1, 2, 3\}$, and that X_1 has been exploded to take values in $\{1, 2, 3, 4, 5, 6\}$ just as in, and using the same probabilities as, Section 4.2.2. If Π_2 consists only of X_1, we have already seen that our search strategy will maximize the information term with a minimum number of thresholds. Assuming now that Π_2 consists of X_1 and some vector $\vec{Y}$ whose components are the other parents of X_2, we have

$$\widehat{I} := \widehat{I}(X_2, \Pi_2) = \widehat{I}(X_2, (X_1, \vec{Y})) = \sum_{ij\vec{k}} \widehat{I}_{ij\vec{k}}$$

where

$$\widehat{I}_{ij\vec{k}} = \widehat{p}(i, j, \vec{k}) \cdot \log \left(\frac{\widehat{p}(i, j, \vec{k})}{\widehat{p}_2(j)\, \widehat{p}_{1Y}(i, \vec{k})} \right).$$

Here, $\widehat{p}(i, j, \vec{k}) = \widehat{P}(X_1 = i, X_2 = j, \vec{Y} = \vec{k})$, $\widehat{p}_2(j) = \widehat{P}(X_2 = j)$, and $\widehat{p}_{1Y}(i, \vec{k}) = \widehat{P}(X_1 = i, \vec{Y} = \vec{k})$. Using $\widetilde{p}$ to denote probabilities after the explosion of X_1, it is easy to see expected relationships such as

$$\widetilde{p}(4, j, \vec{k}) = \frac{4}{7}\widehat{p}(2, j, \vec{k}), \qquad \widetilde{p}_2(j) = \widehat{p}_2(j), \qquad \text{and} \quad \widetilde{p}_{1Y}(2, \vec{k}) = \frac{2}{3}\widehat{p}_{1Y}(1, \vec{k}).$$

Therefore, we can verify, for example, that $\widetilde{I}_{2j\vec{k}} = (2/3)\widehat{I}_{1j\vec{k}}$, and that the overall information terms $\widetilde{I}$ and $\widehat{I}$ are equal. We can also verify, using the log-sum inequality, that after removal of the rth threshold we have

$$I^{(r)}_{r,j,\vec{k}} \leq \widetilde{I}_{r,j,\vec{k}} + \widetilde{I}_{r+1,j,\vec{k}},$$

with equality if and only if

$$\widetilde{P}(X_2 = j | X_1 = r, \vec{Y} = \vec{k}) = \widetilde{P}(X_2 = j | X_1 = r + 1, \vec{Y} = \vec{k}). \tag{4.8}$$

Summing over all appropriate values for indices, we get that $I^{(r)}$, the information term between X_2 and its parents after removal of the rth threshold for X_1, is less than or equal to $\widetilde{I}$, with equality if and only if (4.8) holds for all j and $\vec{k}$. When $r = 1$, for example, both sides of (4.8) are equal to $\widehat{P}(X_2 = j|X_1 = 1, \vec{Y} = \vec{k})$, indicating that the first threshold should be removed. When $r = 2$, the left side is equal to $\widehat{P}(X_2 = j|X_1 = 2, \vec{Y} = \vec{k})$ and the right side is equal to $\widehat{P}(X_2 = j|X_1 = 3, \vec{Y} = \vec{k})$, so we do not have (4.8) and hence a decrease in information. Thus, we would not remove the second threshold.

$\widetilde{I}$, the information with all thresholds in place, is the maximal information among all possible discretizations. If we remove each threshold that leaves the information unchanged, we will still have the maximal information with a minimum number of thresholds. Thus, the top-down search strategy will give the optimal discretization.

Considering all terms in (4.2) independently may result in different discretizations. For example, consider graph 8 from Table 2.2. In this case, (4.2) becomes $\widehat{I}(X_1, X_2) + \widehat{I}(X_1, X_3)$. If $\widehat{P}(X_2 = j|X_1 = 1) = \widehat{P}(X_2 = j|X_1 = 2)$ for all j but $\widehat{P}(X_3 = j|X_1 = 1) \neq \widehat{P}(X_3 = j|X_1 = 2)$ for some j, removing the threshold between 1 and 2 for the discretization of X_1 would leave the information between X_1 and X_2 unchanged but would decrease he information between X_1 and X_3. However, as both information terms, and hence their sum, are maximized with the full discretization for X_1, the top down-search strategy will not allow us to remove the threshold between 1 and 2. That is, it will only remove thresholds between values of X_1 that are indistinguishable in terms of the conditional distributions involving all nodes to connected to X_1.

Chapter 5

CONCLUSIONS

5.1 Conclusion

Over the past decades, Bayesian networks have become a popular representation for encoding uncertainty and extensively employed in areas such as bioinformatics, artificial intelligence, medical diagnostic, and risk management. In these applications, the recovery of the structure of a network is often based on the assumption of discrete or continuous but Gaussian data. For general continuous data, discretization is usually employed but often destroys the very structure one is out to recover. Friedman and Goldszmidt [11] suggest an approach based on the minimum description length principle that chooses a discretization which preserves the information in the original data set, however it is one which is difficult, if not impossible, to implement for even moderately sized networks. This study provides a extremely efficient search strategy which allows one to use the Friedman and Goldszmidt in practice.

In Chapter 4, we show, in the case of ideal "exploded" data where there is a "correct" discretization, that the minimum description length scoring mechanism of Friedman and Goldszmidt will in fact recover the discretization. Just as importantly, we have seen that we can find it from among $2^{m_1 - 1}$ possibilities by making only $m_1 - 1$ comparisons.

In the case of discrete data where superfluous values were not manufactured, for example the two-node network where X_1 is a parent to X_2 that originally takes values in $\{1, 2, 3, 4, 5, 6\}$ and $\{1, 2, 3\}$, respectively, we should aggregate 1 and 2 for node 1 into a single value if $P(X_2 = j|X_1 = 1) = P(X_2 = j|X_1 = 2)$ for all $j \in \{1, 2, 3\}$. From the data, we will only get to see that $\widehat{P}(X_2 = j|X_1 = 1) \approx \widehat{P}(X_2 = j|X_1 = 2)$. Even with a large sample size, because of the approximation, we would still see some decrease in overall information when removing the threshold, so it remains to determine when such a decrease is significant. We have much empirical evidence that we will still be able to recover the correct discretization by comparing DL_{local} for only single threshold removals to DL_{local} for the full discretization. For example, we simulated $100,000$ values for X_1 and X_2 in the two-node network by simulating X_1 in $\{1,2,3,4,5,6\}$ directly (as opposed to first simulating them in $\{1,2,3\}$ and then exploding the data). We chose parameters such that $P(X_2 = j|X_1 = 1) = P(X_2 = j|X_1 = 2)$ and $P(X_2 = j|X_1 = 5) = P(X_2 = j|X_1 = 6)$ for all $j \in \{1,2,3\}$. In Table 5.1, we show the full discretization score, all single threshold removal scores, and the true discretization score. The two incorrect threshold removals (between 1 and 2 and between 5 and 6) stand out as having different DL_{local} values than the rest.

Table 5.1: Local Description Length Score

Discretization	DL local
1|2|3|4|5|6	-29841.52
12|3|4|5|6	-29870.53
1|23|4|5|6	-24456.07
1|2|34|5|6	-29866.02
1|2|3|45|6	-29896.74
1|2|3|4|56	-24585.90
12|345|6	-29929.17

5.2 Limitations and Future Work

While this study provides a significant contribution to the body of knowledge by introducing a efficient search strategy which allows one to use the Friedman and Goldszmidt's discretization method in practice, there are several limitations that warrant future work discussed below.

- For truly continuous data, in principle the minimum description length discretization of Friedman and Goldszmidt and our single threshold search strategy will still work if there is a "true discretization". For the two node example, this would mean that $f_{X_2|X_1}(x|r) = f_{X_2|X_1}(x|s)$ for all r and s in a given interval. (Here, $f_{X_2|X_1}$ is the conditional density of X_2 given X_1.) However, even in this ideal case, it will be difficult to find the true discretization without accurate estimation of mutual information which would now be an integral. We refer the reader to [7] for a numerical estimation technique that outperforms standard binning methods. However, for continuous data where there is no true discretization in terms of the underlying probability density functions, there still may be a discretization that will recover the correct network when using a scores such as AIC, BIC, or MDL. Removal of thresholds in this case will always result in a decrease of information, so we are again faced with having to characterize how significant these changes are in terms of their effect on the recovered network structure. More work is needed to address this issue.

- This study illustrated the single threshold search strategy using a two-node network. While the two-node network single threshold removal principle could use for more complicated networks, future work is required for generalized problem with n-node. When the data contains a large number of values for a continuous variable this procedure may become challenging and expensive. We can use Monte Carlo methods (i.e, Metropolis-Hastings Algorithm, Markov chain Monte Carlo, or stochastic hill-climbing) to overcome this challenge.

- Finally, the future work could use the findings from this study to model real-world phenomena decision making such as medicine and industrial control areas in which variables often have continuous values. For example, the future work could apply the single threshold search strategy presented in this thesis to model the recovery of a genetic regulatory pathway. The nodes of a Bayesian network represent gene expression levels associated with particular genes and the arrows between nodes represent interactions between genes. Gene expression levels are given by continuous data and discretization allows us to think of the random variables for each node as multinomial random variables . The quantification of this discretization is an interesting research problem.

Bibliography

[1] H. Akaike. Information theory and an extension of the maximum likelihood principle. In B.N. Petrov and F. Csaki, editors, Second international symposium on information theory, pages 267–281. Budapest: Academiai Kiado, 1973.

[2] H. Akaike. A new look at the statistical model identification. IEEE Transactions on Automatic Control, 19(6):716–723, 1974.

[3] H. Akaike. Likelihood of a model and information criteria. Journal of Econometrics, 16:3–14, 1981.

[4] H. Bozdogan. Akaike's information criterion and recent developments in information complexity. Journal of Mathematical Psychology, 44:62–91, 2000.

[5] T. Chen, H.L. He, and G.M. Church. Modeling gene expression with differential equations. In Pacific Symposium Biocomputing '99, pages 29–40. 1999.

[6] T.M. Cover and J.A. Thomas. Elements of Information Theory. John Wiley & Sons, Hoboken, New Jersey, 2006.

[7] Carsten O Daub, Ralf Steuer, Joachim Selbig, and Sebastian Kloska. Estimating mutual information using b-spline functions–an improved similarity measure for analysing gene expression data. BMC bioinformatics, 5(1):118, 2004.

[8] A. Dobra, C. Hans, B. Jones, J.R. Nevins, G. Yao, and M. West. Sparse graphical models for exploring gene expression data. Journal of Multivariate Analysis, 90(1):196–212, 2004.

[9] N. Dojer, A. Gambin, A Mizera, B. Wilczynski, and J. Tiuryn. Applying dynamic Bayesian networks to perturbed gene expression data. BMC Bioinformatics, 7:249–260, 2006.

[10] N. Fenton and M. Neil. Risk Assessment and Decision Analysis with Bayesian Networks. CRC Press, Taylor & Francis Group, Boca Raton, FL, 2013.

[11] N. Friedman and M. Goldszmidt. Discretizing continuous attributes while learning Bayesian networks. In Proceedings of ICML-1996, pages 157–165.

[12] N. Friedman and D. Koller. Being Bayesian about network structure. Machine Learning, 50:95–126, 2003.

[13] N. Friedman, M. Linial, I. Nachman, and D. Pe‘er. Using Bayesian networks to analyze expression data. Journal of Computational Biology, 7:601–620, 2000.

[14] D. Heckerman. A tutorial on learning with Bayesian networks. Technical report, Microsoft Corporation, 1995.

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

[16] D. Husmeier. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. Bioinformatics, 19(17):2271–2282, 2003.

[17] D. Husmeier. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic bayesian networks. Bioinformatics, 19(17):2271–2282, 2003.

[18] S. Imoto, S. Kim, T. Goto, S. Miyano, S. Aburatani, K. Tashiro, and S. Kuhara. Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network. Journal of Bioinformatics and Computational Biology, 1:231–252, 2003.

[19] E.D. Jarvis, V.A. Smith, K. Wada, M.V. Rivas, M. McElroy, T.V. Smulders, P. Carninci, Y. Hayashizaki, F. Dietrich, X. Wu, P. McConnell, J. Yu, P.P. Wang, A.J. Hartemink, and S. Lin. A framework for integrating the Songbird brain. Journal of Comparative Physiology A, 188:961–980, 2002.

[20] R.L. Kashvap. A Bayesian comparison of different classes of dynamic models using empirical data. IEEE Transactions on Automatic Control, 22(5):715–727, 1977.

[21] I.M. Ong, J.D. Glasner, and D. Page. Modeling regulatory pathways in E. coli from time series expression profiles. Bioinformatics, 18:241–248, 2002.

[22] J. Pearl. Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufman, San Francisco, CA, 1988.

[23] J. Rissanen. Modeling by shortest data description. Automatica, 14(5):465–658, 1978.

[24] G. Schwartz. Estimating the dimension of a model. The Annals of Statistics, 5(2):461–464, 1978.

[25] J. Wang. Recovering Bayesian Networks Through MCMC with Applications to Gene Regulatory Networks. PhD dissertation, University of Colorado at Boulder, Department of Applied Mathematics, August 2007.

[26] D.E. Zak, F.J. Doyle, G.E. Goyne, and J.S. Schwaber. Simulation studies for the identification of genetic networks from cDNA array and regulatory activity data. Proc. 2nd Intl. Conf. Systems Biology, pages 231–238, 2001.