2010

Bayesian model averaging using k-best bayesian network structures

Lavanya Ram
Iowa State University

Bayesian model averaging using k-best bayesian network structures

by

Lavanya Ram

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Science

Program of Study Committee:
Jin Tian, Major Professor
Pavankumar R Aduri
Vasant G Honavar

Iowa State University

Ames, Iowa

2010

DEDICATION

I would like to dedicate this thesis to Lord Almighty for the blessings and to my parents Ram and Radha for their unconditional support and encouragement throughout my work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I take this opportunity to thank everyone who has helped me directly or indirectly throughout my Master's degree and during the course of my thesis. However small the help might have been, I want to stress that it was appreciated more than you know.

First, I would like to thank my advisor, Dr.Jin Tian for helping me gain an insight into the research problem and guiding me in every aspect starting from the basics to the finer details of my thesis. Dr.Tian's constant encouragement, patience and support has been invaluable during the course of my Master's degree. I would like to thank Dr.Jin Tian for his extreme patience during the learning (and still learning) phases of my Graduate study.

I would also like to thank my committee members - Dr.Vasant Honavar and Dr.Pavan Aduri for taking the time to review my thesis and giving suggestions for improvement. I would like to thank Linda Dutton for helping me with all the paper work and making it easier at every stage of my Graduate study.

I would like to thank Iowa State University for providing us with state of the art facilities and resources to make ISU one of the best places to conduct research in Machine Learning.

I would like to thank many of my friends for their many hours of debugging my code and for sharing their experiences in making me a better programmer. I would like to thank Yetian Chen and Ru He for their helpful discussions. I would like to thank my roommates - Samyukta Soothram and Alakananda Mysore for the wonderful evenings of fun and great food. I would like to thank all my other friends who stood by me during difficult times giving me the right mix of fun, advice and memories and a wonderful two years in Ames.

Last but not the least, I would like to thank my parents - Ram and Radha for their continuous patience, understanding, advice, encouragement and support during my Graduate

education. Their encouragement has played an inevitable role in pushing me to higher levels and has always made my aspirations higher and never ending.

ABSTRACT

Bayesian networks are being widely used in various data mining tasks for probabilistic inference and causual modeling [Pearl (2000), Spirtes *et al.* (2001)]. Learning the best Bayesian network structure is known to be NP-hard [Chickering (1996)]. Also, learning the single best Bayesian network structure does not always give a good approximation of the actual underlying structure. This is because in many domains, the number of high-scoring models is usually large.

In this thesis, we propose that learning the top-k Bayesian network structures and model averaging over these k networks gives a better approximation of the underlying model. The posterior probability of any hypotheses of interest is computed by averaging over the top-k Bayesian network models. The proposed techniques are applied to flow cytometric data to make causal inferences in human cellular signaling networks. The causal inferences made about the human T-cell protein signaling model by this method is compared with inferences made by various other learning techniques which were proposed earlier [Sachs *et al.* (2005), Koch *et al.* (2009)]. We also study and compare the classification accuracy of the top-k networks to that of the single MAP network.

In summary, this thesis describes:

1. Algorithm for learning the top-k Bayesian network structures.
2. Model averaging based on the top-k networks.
3. Experimental results on the posterior probabilities of the top-k networks.
4. How the top-k Bayesian networks can be applied to learn protein signaling networks with Results of top-k model averaging on the CYTO data.
5. Results of Classification Accuracy of the top-k networks.

CHAPTER 1. OVERVIEW AND MOTIVATION

This chapter provides an introduction to the basics and the motivation behind the thesis. A brief overview of the contributions and an outline of the overall structure of the thesis is also provided.

1.1 Introduction

There are many data mining tasks in which we would like to study the influence of one feature on another feature in a data domain. For example, suppose that there are two events which could cause grass to be wet: either the sprinkler is on, or it is raining. Also, suppose that the rain has a direct effect on the use of the sprinkler (namely that when it rains, the sprinkler is usually not turned on)[Russell and Norvig (2003)]. Probabilistic inferences in these situations have been made using the Bayes theorem.

1.1.1 Bayes' Theorem

Bayes' theorem relates the conditional and prior probabilities of two events A and B, provided that the probability of B does not equal zero:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \tag{1.1}$$

where:

- $P(A)$ is the prior probability of A. It is "prior" in the sense that it does not take into account any information about B.

- $P(A|B)$ is the conditional probability of A, given B. It is also called the posterior probability because it is derived from or depends upon the specified value of B.
- $P(B|A)$ is the conditional probability of B given A. It is also called the likelihood.
- $P(B)$ is the prior or marginal probability of B, and acts as a normalizing constant.

Bayes' theorem gives a mathematical representation of how $P(A|B)$ is related to the converse conditional probability $P(B|A)$.

1.1.2 Conditional Independence

Two events A and B are conditionally independent given a third event C precisely if the occurrence or non-occurrence of A and B are independent events in their conditional probability distribution given C. In the standard notation of probability theory,

If A is conditionally independent of B given C:

$$P(A|B,C) = P(A|C) \tag{1.2}$$

This conditional independence can be represented as:

$$I_P(A,B|C) \tag{1.3}$$

1.1.3 Computing Joint Probability Distribution using Bayes' Rule

The joint ptobability distribution of two events A and B is given by P(A,B). Based on the Bayes' Rule this can be computed as:

$$P(A \cap B) = P(A) * P(B|A) \tag{1.4}$$

In situations where we need to make inferences with only two variables in our domain, Bayes rule can be easily used along with the rules of probability to make probabilistic inferences. However, when the number of variables in the domain increases, the probabilistic inferences become less straightforward owing to the complexity of the probabilistic relationships between the various events and the number of terms that need to be summed to calculate the joint probabilities. The model starts becoming less intuitive and very complex.

1.2 Bayesian Networks as a Modeling Framework for Large Instances

Bayesian networks address the problems of 1) representing the joint probability distribution of a large number of random variables and 2) doing Bayesian inference with these variables.

The probabilistic relationships can be modeled easily and more intuitively using a Bayesian network as shown in Figure 1.1 for the sprinkler example we discussed above.

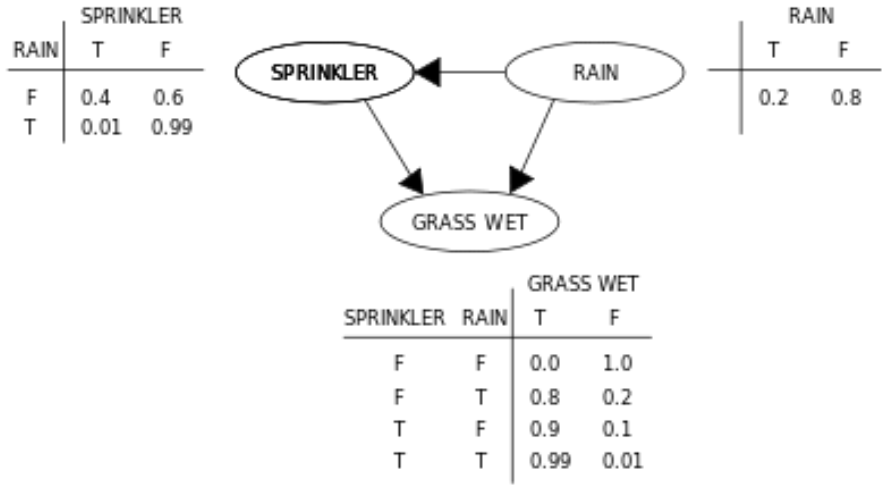

Figure 1.1 A Bayesian network with conditional probability tables

The three variables in our case are: R (for Rain), S(for Sprinkler) and G(for Grass Wet). Each of the three variables have two possible values - T (for true) and F (for false).

Before we formally define a Bayesian network, it is useful to define the terminology we will be using for graphs. We will do this in the following subsection.

1.2.1 Graph Theory Basics

Definition 1 *[Neapolitan (2003)] defines a* ***Directed Graph*** *as follows:*

A Directed graph is a pair (V, E)*, where* V *is a finite nonempty set whose elements are called* ***nodes*** *(or vertices), and* E *is a set of ordered pairs of distinct elements of* V*. Elements of* E *are called* ***edges*** *(or arcs), and if* $(X, Y) \in E$*, we say there is an edge from* X *to* Y *and that* X *and* Y *are each* ***incident*** *to the edge. If there is an edge from* X *to* Y *or from* Y *to* X*, we say* X *and* Y *are* ***adjacent****.*

Definition 2 *[Neapolitan (2003)] defines a* ***Path*** *in a Directed graph as follows:*

Suppose we have a set of nodes $[X_1, X_2, ...X_k]$*, where* $k \geq 2$*, such that* $(X_{i1}, X_i) \in E$ *for* $2 \leq i \leq k$*. We call the set of edges connecting the* k *nodes as a* ***path*** *from* X_1 *to* X_k *. The nodes* $X_2, ...X_{k1}$ *are called* ***interior nodes*** *on path* $[X_1, X_2, ...X_k]$*. The* ***subpath*** *of path* $[X_1, X_2, ...X_k]$ *from* X_i *to* X_j *is the path* $[X_i, X_{i+1}, ...X_j]$ *where* $1 \leq i < j \leq k$*.*

Definition 3 *[Neapolitan (2003)] defines a* ***directed cycle*** *as a path from a node to itself.*

Definition 4 *[Neapolitan (2003)] defines a* ***simple path*** *as a path containing no subpaths which are directed cycles.*

Definition 5 *[Neapolitan (2003)] defines a* ***DAG*** *as follows:*

A directed graph G *is called a* ***directed acyclic graph (DAG)*** *if it contains no directed cycles. Given a DAG* $G = (V, E)$ *and nodes* X *and* Y *in* V*,* Y *is called a* ***parent*** *of* X *if there is an edge from* Y *to* X*,* Y *is called a* ***descendent*** *of* X *and* X *is called an* ***ancestor*** *of* Y *if there is a path from* X *to* Y *, and* Y *is called a* ***nondescendent*** *of* X *if* Y *is not a descendent of* X*.*

Definition 6 *[Neapolitan (2003)] Suppose we have a joint probability distribution* P *of the random variables in some set* V *and a DAG* $G = (V, E)$*. We say that* (G, P) *satisfies the* ***Markov condition*** *if for each variable* $X \in V, \{X\}$ *is conditionally independent of the set of all its nondescendents given the set of all its parents. This means if we denote the sets of parents and nondescendents of* X *by* PA_X *and* ND_X *respectively, then* $I_P(\{X\}, ND_X|PA_X)$*. When* (G, P) *satisfies the Markov condition, we say* G *and* P *satisfy the Markov condition with each other.*

If X is a root, then its parent set PA_X is empty. So in this case the Markov condition means $\{X\}$ is independent of ND_X . That is, $I_P(\{X\}, ND_X)$. Also, $I_P(\{X\}, ND_X|PA_X)$ implies $I_P(\{X\}, B|PA_X)$ for any $B \subseteq ND_X$. For example, the independence relations in the DAG in Figure 1.2 are:

1. $I_P(\{C\}, \{H, B, F\}|\{L\})$

2. $I_P(\{B\}, \{L, C\}|\{H\})$

3. $I_P(\{F\}, \{H, C\}|\{B, L\})$

4. $I_P(\{L\}, \{B\}|\{H\})$

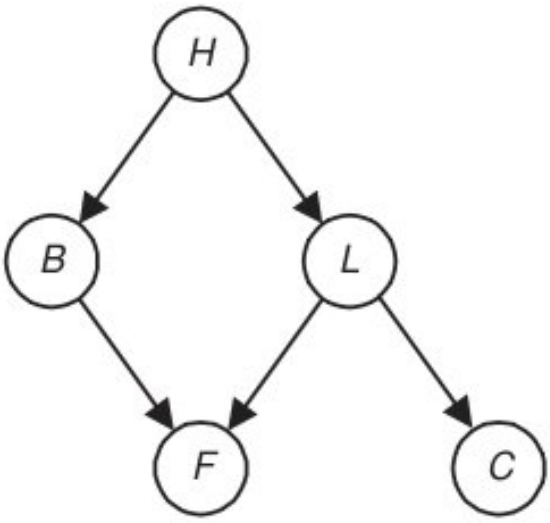

Figure 1.2 A DAG illustrating the Markov condition

If (G, P) satisfies the Markov condition, then P is equal to the product of its conditional distributions of all nodes given values of their parents, whenever these conditional distributions exist.

Now, based on Markov's joint probability distribution, we can revisit the example in Figure 1.1. The joint probability function can be represented as:

$$P(G, S, R) = P(G|S, R)P(S|R)P(R) \tag{1.5}$$

Definition 7 *[Neapolitan (2003)] defines a* ***Bayesian Network*** *as follows:*

Let P be a joint probability distribution of the random variables in some set V, and $G = (V, E)$ be a DAG. We call (G, P) a Bayesian network if (G, P) satisfies the Markov condition.

1.3 Bayesian Networks in Data Mining

Bayesian networks are being widely used in various data mining tasks for probabilistic inference and causal modeling [Pearl (2000), Spirtes *et al.* (2001)]. In the Bayesian approach, we provide a prior probability distribution over the space of possible Bayesian networks and then compute the posterior distributions $P(G|D)$ of the network structure G given data D. We can then compute the posterior probability of any hypotheses of interest by averaging over all possible networks. In some applications, we are interested in structural features. For example, in causal discovery, we are interested in the causal relations among variables, represented by the edges in the network structure [Heckerman *et al.* (1999)]. In other applications, we are interested in predicting the posterior probabilities of new observations, for example, in classification problems.

1.4 Inherent problems in the application of Bayesian networks

Bayesian networks have been known to model noisy and unequally distributed data with good confidence. However, One major challenge in the practical application of Bayesian networks is the learning of the Bayesian network structures from data. The number of possible network structures is superexponential $O(n!2^{n(n-1)/2})$ in the number of variables n. For example, there are about 10^4 directed acyclic graphs (DAGs) on 5 nodes, and 10^{18} DAGs on 10 nodes. As a result, it is impractical to sum over all the possible structures other than for tiny domains (less than 7 variables).

1.5 Motivation behind the Thesis

The most common solution to overcome some of the problems discussed above is to use model selection approach in which we use the relative posterior probability $P(D, G)$ (or other measures) as a scoring metric and then attempt to find a single network with the best score, the MAP network. We then use that model (or its Markov equivalence class) to make future predictions. This may be a good approximation if the amount of data is large relative to the

size of the model such that the posterior is sharply peaked around the MAP model. However, in domains where the amount of data is small relative to the size of the model there are often many high-scoring models with non-negligible posterior. In this situation, using a single model could lead to unwarranted conclusions about the structure features and also poor predictions about new observations. For example, the edges that appear in the MAP model do not necessarily appear in other approximately equally likely models.

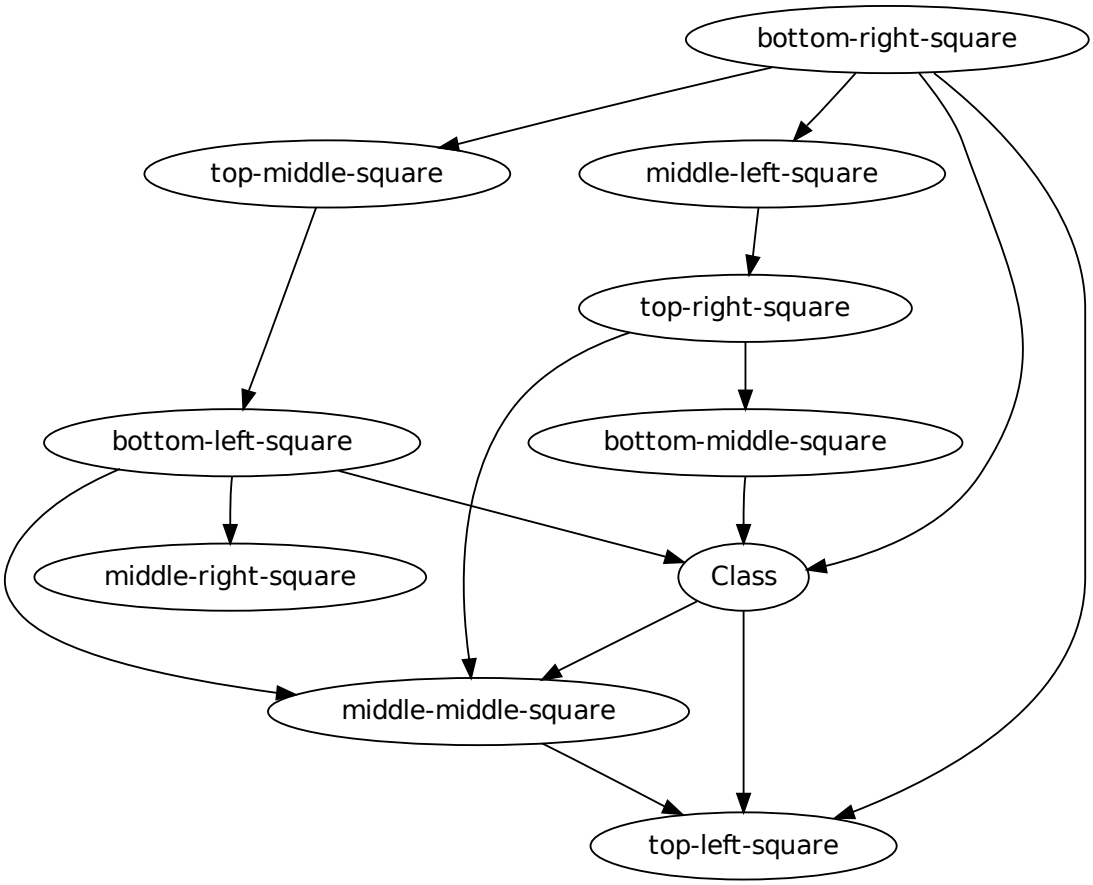

Figure 1.3 Top1 : Top network for the Tic-tac-toe Dataset with $P(G|D)$ = 0.011193

Consider two of the 76 highest scoring Bayesian networks for the tic-tac-toe Dataset [Asuncion and Newman (2007)] shown in Figure 1.3, 1.4. The Top1 network and the Top63 network

both have a BDe metric score of -9423.068333 and posterior probabilities of 0.011193. Details about the BDe metric can be found in [Heckerman *et al.* (1995)]. It can be observed that the two structures differ in their skeleton structures and are therefore from different equivalence classes although they have the same BDe score. If only the single top network were considered in making inferences, we might make unwarranted conclusions during structure learning and also poor predictions in classification. In the tic-tac-toe dataset case, there are 76 high scoring models. However, there are various datasets where the number of high scoring models is large. In such a case, the impact of choosing the single best model over model averaging to make inferences could be more substantial. This is the main motivation behind this thesis.

Also, the model selection may be sensitive to the data samples given in the sense that a different set of data (from the same distribution) might well lead to a different MAP model. In such cases, using Bayesian model averaging is preferred.

1.5.1 Bayesian Model Averaging for Protein Signaling Networks

Cell signaling is part of a complex system of communication that governs basic cellular activities and coordinates cell actions [Günther Witzany (2010)]. The ability of cells to perceive and correctly respond to their microenvironment is the basis of development, tissue repair, and immunity as well as normal tissue homeostasis. Errors in cellular information processing are responsible for diseases such as cancer, autoimmunity, and diabetes. By understanding cell signaling, diseases may be treated effectively and, theoretically, artificial tissues may be yielded.

Due to the noisy and probabilistic nature of biological signaling data, Bayesian networks have often been considered a good means to learn the causal network behind them. We will study the results of the model averaging of the top-k Bayesian network structures on the CYTO dataset [Sachs *et al.* (2005)] in the Experiments and Results chapter.

1.6 Organization of the Thesis

The following chapters in this thesis are organized as follows:

- In Chapter 2, a study of some of the challenges in Bayesian network learning is presented. An overview of related work in the area is also provided.
- In Chapter 3, the algorithm for finding the top-k networks is presented along with the equations showing how feature probability can be calculated based on the top-k structures.
- In Chapter 4, we study the trend of the Posterior probability of the top-k networks $P(G|D)$ with increase in k-value.
- In Chapter 5, we study how the top-k averaging can be used to predict the protein-signaling network from the CYTO dataset and compares it to previous work in that area.
- In Chapter 6, we study the classification accuracy of the top-k networks.
- In Chapter 7, we conclude the thesis work by summarizing the work done in the thesis.

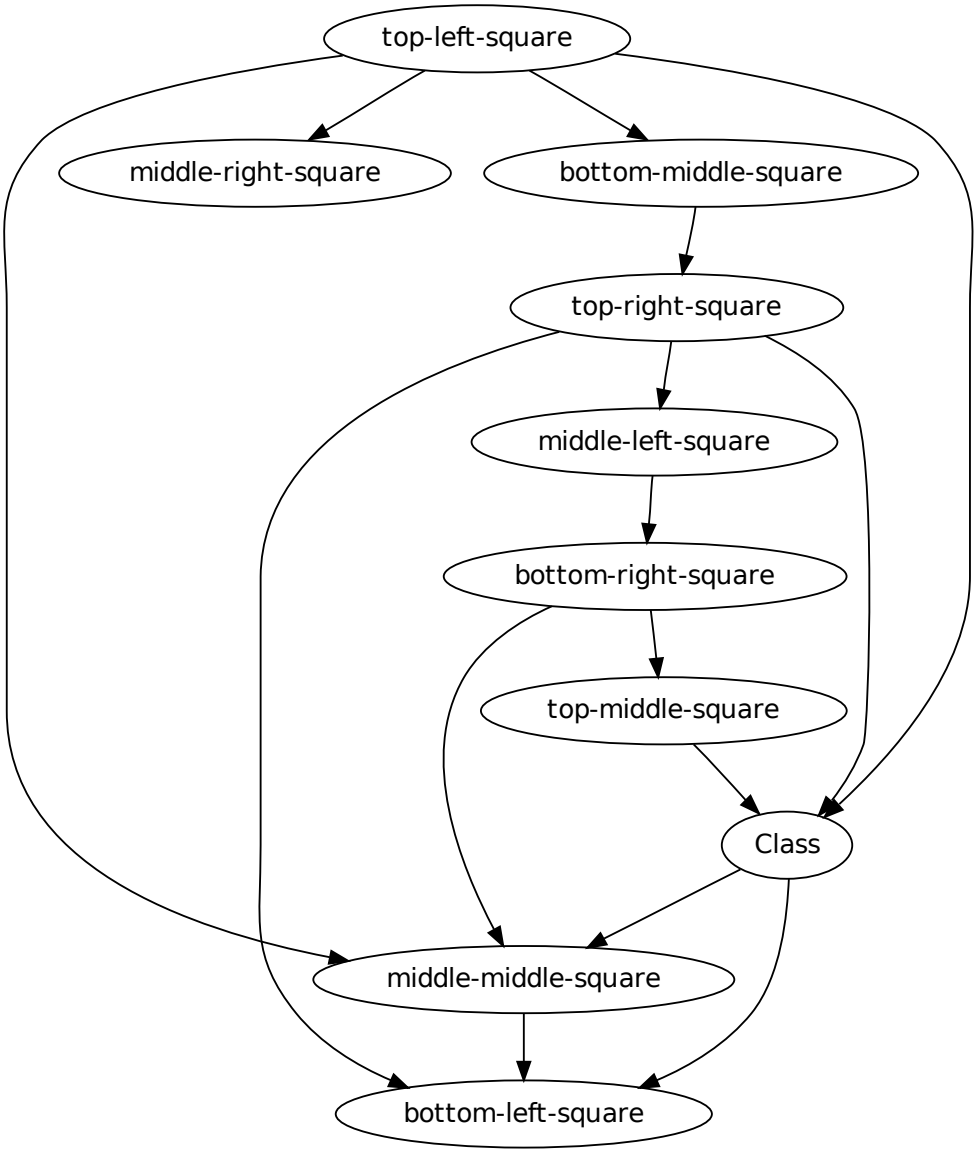

Figure 1.4 Top63: Another top network for the Nursery Dataset with same $P(G|D) = 0.011193$

CHAPTER 2. REVIEW OF LITERATURE

The problem of learning Bayesian network structures from data has been widely studied in the past. In the following sections, we study the major challenges in Bayesian structure learning and summarize some of the earlier work in this area stating areas for improvement in each idea.

2.1 Major challenges in Bayesian networks learning

As we saw earlier, the main problem in the application of Bayesian networks for various data mining tasks is that the finding of the MAP network is NP-hard [Chickering (1996)] and hence a typical approach to search for the MAP structure is to use some sort of heuristic algorithms which may get stuck in a local maxima leading to unwarranted conclusions about the data domain. Also, making inferences based on the single MAP structures also has problems in domains where the amount of data is small relative to the size of the model. In such domains, the posterior may not be sharply peaked around the MAP model. There are usually many high-scoring models with non-negligible posterior. Therefore, relying on the single MAP structure may also lead to unwarranted conclusions about the structure features and poor predictions about new observations. In the following sections, we examine some of the MAP learning and model averaging approaches devised in the past to learn Bayesian network structures and some applications of Bayesian networks to cell signaling networks.

2.2 Computation of exact posterior probabilities

Recently there has been progress on computing exact posterior probabilities of structural features such as edges or subnetworks using dynamic programming techniques [Koivisto and

Sood (2004), Koivisto (2006), Tian and He (2009)]. These techniques have exponential time and memory complexity and are capable of handling datasets with upto around 20 variables. One problem with these algorithms is that they can only compute posteriors over modular features such as directed edges but can not compute non-modular features such as Markov blankets or paths ("is there a path between nodes X and Y").

Another problem is that it is very expensive to perform data prediction tasks. They can compute the exact posterior of new observational data $P(X|D)$ but the algorithms have to be re-run for each new data case x.

2.3 Model averaging techniques in the past

When computing exact posterior probabilities of features is not feasible, one approach that has been proposed is to approximate full Bayesian model averaging by finding a set of high-scoring networks and make prediction using these models [Madigan and Raftery (1994), Heckerman *et al.* (1997)]. This leaves open the question of how to construct the set of representative models. Suggestions were made to use the high scoring models encountered in greedy or beam search as the representative models and make posterior probability calculations based on these models. However, the results of the inference may be biased towards the networks favored by the particular search strategy adopted and may not be representative of the real data. [Madigan and Raftery (1994)] suggests using a set of opening models based on expert knowledge and then follow a greedy search based on the Posterior model probabilities. This technique again requires prior knowledge about the domain and could get stuck in a local maxima during the greedy search resulting in biased inferences. One possible approach is to use the bootstrap technique which has been studied in [Friedman *et al.* (1999)]. However, there are still many questions that need further study on how to use the bootstrap to approximate the Bayesian posterior over features.

One theoretically well-founded approach is to use Markov Chain Monte Carlo (MCMC) techniques. [Madigan and York (1995)] used MCMC algorithm in the space of network structures(i.e., DAGs). [Friedman and Koller (2003)] developed a MCMC procedure in the space

of node orderings which was shown to be more efficient than MCMC in the space of DAGs and to outperform the bootstrap approach in [Friedman *et al.* (1999)] as well. [Eaton and Murphy (2007)] developed a hybrid MCMC method (DP+MCMC) that first uses the dynamic programming technique in [Koivisto (2006)] to develop a global proposal distribution and then runs MCMC in the DAG space. Their experiments showed that DP+MCMC algorithm converged faster than previous two methods [Friedman and Koller (2003), Madigan and York (1995)] and resulted in more accurate structure learning. One common problem to the MCMC and bootstrap approach is that there is no guarantee on the quality of the approximation in finite runs.

[Madigan and Raftery (1994)] has proposed to discard the models whose posterior probability is much lower than the best ones (as well as complex models whose posterior probability is lower than some simpler one). In this thesis, we study the approach of approximating Bayesian model averaging using a set of Bayesian networks. It is intuitive to make predictions using a set of the best models, and we believe it is due to the computational difficulties of actually finding the best networks that this idea has not been systematically studied. In this thesis, an algorithm for finding the k-best network structures is studied by generalizing the dynamic programming algorithm for finding the optimal Bayesian network structures in [Silander and Myllymaki (2006), Singh and Moore (2005)]. We demonstrate the algorithm on several datasets from the UCI Machine Learning Repository [Asuncion and Newman (2007)]. An empirical study is then presented on the quality of Bayesian model averaging using the k-best networks in classification.

CHAPTER 3. MODEL AVERAGING USING TOP-k BAYESIAN NETWORK STRUCTURES

We study the problem of learning bayesian network structures from data. In this Chapter, we study the algorithm for finding the k-best Bayesian network structures using a dynamic programming approach. We propose to compute the posterior probability of any hypotheses of interest by Bayesian model averaging over the k-best Bayesian network structures.

In particular, it is reasonable to use the k-best networks. However, this idea has not been systematically studied due to the computational difficulties of actually finding the k-best networks. In this thesis, we will explore this idea. We developed an algorithm for finding the k-best network structures by generalizing the dynamic programming algorithm for finding the optimal Bayesian network structures in [Singh and Moore (2005), Silander and Myllymaki (2006)].

We applied our algorithm to several data sets from the UCI Machine Learning Repository [Asuncion and Newman, 2007]. We then make some observations on Bayesian averaging using the k-best networks.

3.1 Bayesian Learning of Bayesian Networks

A Bayesian network is a DAG G that encodes a joint probability distribution over a set $X = \{X_1, ..., X_n\}$ of random variables with each node of the graph representing a variable in X. For convenience, we will typically work on the index set $V = \{1, ..., n\}$ and represent a variable X_i by its index i. We use $X_{Pa_i} \subseteq X$ to represent the set of parents of X_i in a DAG G and use $Pa_i \subseteq V$ to represent the corresponding index set.

In the problem of learning BNs from data, we are given a training data set $D = \{x^1, x^2, ..., x^N\}$,

where each x^i is a particular instantiation over the set of variables X. In this paper we only consider situations where the data are complete, that is, every variable in X is assigned a value. In the Bayesian approach to learning Bayesian networks from the training data D, we compute the posterior probability of a network G as

$$P(G|D) = \frac{P(D|G)P(G)}{P(D)} \tag{3.1}$$

Assuming global / local parameter independence, and parameter / structure modularity, $lnP(D|G)P(G)$ can be decomposed into a summation of so-called local scores as [Cooper and Herskovits (1992), Heckerman *et al.* (1995)]

$$lnP(G, D) = \sum_{i=1}^{n} score_i(Pa_i : D) \equiv score(G : D), \tag{3.2}$$

where, with appropriate parameter priors, $score_i(Pa_i : D)$ has a closed form solution. In this thesis, we will focus on discrete random variables assuming that each variable X_i can take values from a finite domain. We will use the popular BDe score for $score_i(Pa_i : D)$ and we refer to [Heckerman *et al.* (1995)] for its detailed expression. Often for convenience we will omit mentioning D explicitly and use $score_i(Pa_i)$ and $score(G)$.

In the Bayesian framework, we compute the posterior probability of any hypothesis of interest h by averaging over all possible networks.

$$P(h|D) = \sum_{G} P(h|G, D)P(G|D). \tag{3.3}$$

Since the number of possible DAGs is superexponential in the number of variables n, it is impractical to sum over all DAGs unless for very small networks. One solution is to approximate this exhaustive enumeration by using a selected set of models in $\mathcal{G}$.

$$\hat{P}(h|D) = \frac{\sum\limits_{G\in\mathcal{G}} P(h|G, D)P(G|D)}{\sum\limits_{G\in\mathcal{G}} P(G|D)} \tag{3.4}$$

$$= \frac{\sum\limits_{G\in\mathcal{G}} P(h|G, D)P(G, D)}{\sum\limits_{G\in\mathcal{G}} P(G, D)} \tag{3.5}$$

where $\hat{P}(.)$ denotes approximated probabilities. In the model selection approach, we find a high-scoring model G_s and use it to make predictions:

$$\hat{P}(h|D) = P(h|G_s, D) \tag{3.6}$$

In this thesis, we will perform model averaging using the set $\mathcal{G}$ of k-best networks.

We can estimate the posterior probabilities of a network $G \in \mathcal{G}$ as

$$\hat{P}(G|D) = \frac{P(G, D)}{\sum_{G \in \mathcal{G}} P(G, D)} \tag{3.7}$$

We can then estimate the posterior probability of hypothesis h by

$$\hat{P}(h|D) = \sum_{G \in \mathcal{G}} P(h|G, D)\hat{P}(G|D) \tag{3.8}$$

If we are interested in computing the posteriors of structural features such as edges, paths, Markov Blankets, etc., let $\mathcal{F}$ be a structural feature represented by an indicator function such that $\mathcal{F}(G)$ is 1 if the feature is present in G and 0 otherwise. We have $P(\mathcal{F}|G, D) = \mathcal{F}(G)$ and

$$\hat{P}(\mathcal{F}|D) = \sum_{G \in \mathcal{G}} \mathcal{F}(G)\hat{P}(G|D) \tag{3.9}$$

If we are interested in predicting the posteriors of future observations, let D^T be a set of new data examples independent of D. Then

$$\hat{P}(D^T|D) = \sum_{G \in \mathcal{G}} P(D^T|G, D)\hat{P}(G|D) \tag{3.10}$$

where

$$lnP(D^T|G, D) = \frac{lnP(D^T, D|G)}{P(D|G)} \tag{3.11}$$

$$= score(G : D^T, D) - score(G : D) \tag{3.12}$$

3.2 Learning the top-k structures

We find the k-best structures using the dynamic programming techniques extending the algorithm for finding the optimal Bayesian network structures in [Silander and Myllymaki (2006)]. Our algorithm consists of three steps:

1. Compute the local scores for all possible $n2^{n-1}(i, Pa_i)$ pairs.
2. For each variable $i \in V$, find the k-best parent sets in parent candidate set C for all $C \subseteq V \backslash \{i\}$.
3. Find the k-best networks.

3.2.1 Computing Local Scores

Computing local scores is exactly the same as in [Silander and Myllymaki (2006)] and we will use their algorithm. Assuming that we have calculated all the local scores, next we describe how to accomplish Steps 2 and 3 using dynamic programming technique.

3.2.2 Finding the k-best parent sets

We can find the k-best parent sets for a variable v from a candidate set C recursively. The k-best parent sets in C for v are the k-best sets among the whole candidate set C itself, and the k-best parent sets for v from each of the smaller candidate sets $\{C \backslash \{c\} | c \in C\}$. Therefore, to compute the k-best parent sets for v for every candidate set $C \subseteq V \backslash \{v\}$, we start with sets of size $|C| = 1$, then consider sets of $|C| = 2$, and so on, until the set $C = V \backslash \{v\}$.

The skeleton algorithm for finding the k-best parent sets for v from a candidate set C is given in Algorithm 1, where we use $bestParents_v[S]$ to denote the k best parent sets for variable v from candidate set S stored in a priority queue, and the operation $Merge(.,.)$ outputs a priority queue of the k best parents among the input two priority queues of k elements. Assuming that the merge operation takes time $T(k)$, finding the k-best parent sets for v from a candidate set C takes time $O(T(k) * |C|)$, and computing for all $C \subseteq V \backslash \{v\}$ takes time:

$\sum_{|C|=1}^{n-1} T(k) * |C| * \binom{n-1}{|C|} =$

$O(T(k)(n-1)2^{n-2})$.

Algorithm 1 Finding the k-best parent sets for variable v from a candidate set C

Input:

$score_v(C)$: Local scores

$bestParents_v[S]$: priority queues of the k-best parent sets for variable v from candidate set S for all $S \subseteq C$ with $|S| = |C| - 1$

Output:

$bestParents_v[C]$: a priority queue of the k-best parents of v from the candidate set C

---

for all $S \subseteq C$ such that $|S| = |C| - 1$ **do**

{

$bestParents_v[C] \leftarrow Merge(bestParents_v[C], bestParents_v[S])$

}

Insert C into $bestParents_v[C]$ if $score_v(C)$ is larger than the minimum score in $bestParents_v[C]$

end for

3.2.3 Finding the k-best network structures

Having calculated the k-best parent sets for each variable v from any set C, finding the k-best network structures over a variable set W can be done recursively. We will exploit the fact that every DAG has a *sink*, a node that has no outgoing edges. First for each variable $s \in W$, we can find the k-best networks over W with s as a sink. Then the k-best networks over W are the k-best networks among $\{$the k-best networks over W with s as a sink : $s \in W\}$.

The k-best networks over W with s as a sink can be identified by looking at the k-best parent sets for s from the set $W \backslash \{s\}$ and the k-best networks over $W \backslash \{s\}$.

More formally, let $bestParents_s[C][i]$ denote the ith best parent set for variable s in the candidate parent set C. Let $bestNets[W][j]$ denote the jth best network over W. Then the k-best networks over W with s as a sink can be identified by finding the k-best scores among

$$\{score_s(bestParents_s[W\backslash\{s\}][i]) + score(bestNets[W\backslash\{s\}][j]) : i = 1, ..., k, j = 1, ..., k\}.$$

This can be done by using a standard best-first graph search algorithm.

The skeleton algorithm for finding the k-best network structures over a set W is given in Algorithm 2. Let the time spent on the best-first search be $T'(k)$. In the worst case, all k^2 nodes may need to be visited. The complexity of finding the k-best network structures is

$$\sum_{|W|=1}^{n} T'(k) * |W| * \binom{n}{|W|} = O(T'(k)n2^{n-1}).$$

Algorithm 2 Finding the k-best network structures over set W

Input:

$bestParents_i[S]$: priority queues of the k-best parent sets for each variable $i \in V$ from any candidate set $S \subseteq V - \{i\}$

$bestNets[S]$: priority queues of the k-best network structures over all $S \subseteq W$ with $|S| = |W| - 1$

Output:

$bestNets[W]$: a priority queue of the k-best networks over W

for all $s \in W$ **do**
{
Consider search space $\{(i, j) : i = 1, ..., k, j = 1, ..., k\}$ with root node (1,1), children of (i, j) being $(i + 1, j)$ and $(i, j + 1)$, and the value of each node given by
$value(i, j) = score_s(bestParents_s[W\backslash\{s\}][i]) + score(bestNets[W\backslash\{s\}][j])$
repeat
a best-first graph search over the space $\{(i, j)\}$
until $value(i, j) < score(bestNets[W][k])$ {
Construct a BN G from the network $bestNets[W\backslash\{s\}][j]$ and setting the set $bestParents_s[W\backslash\{s\}][i]$ as the parents of s
Insert the network G into $bestNets[W]$ if G is not in the queue already
} }
end for

3.3 Computing posterior probability of structural features

After learning the top-k structures, we are interested in knowing the posterior probabilities of the structural features in the DAGs to make causal inferences. Here, we are interested in finding two posterior quantities:

1. Posterior probability of a particular DAG given the data $P(G|D)$
2. Posterior probability of a feature in the graph given the data (say, for example, an edge) $P(f|D)$

Since we are using Bayesian score to score our networks, we have

$$score(G:D) = lnP(G,D) \tag{3.13}$$

We are interested in the calculation of $P(G|D)$ which is given by the following equation:

$$P(G|D) = \frac{P(G,D)}{\sum_G P(G,D)} \tag{3.14}$$

Let f be a structural feature such that $f(G)$ is 1 if the feature is present in G and 0 otherwise. Given a dataset D, we are interested in computing the posterior $P(f|D)$ which is given by:

$$P(f|D) = \sum_G f(G)P(G|D) \tag{3.15}$$

3.3.1 Computing approximate posteriors of structural features from top-k networks

In this section, we examine how we can approximately calculate the posteriors of the structural features from the top-k networks. Let us denote the top-k best scored networks we computed by $F = \{G|score(G:D)$ is among top-k best$\}$. Let F' denote the rest of the possible DAGs. We approximate the posterior distributions using only the top-k best scored networks as:

$$\hat{P}(G|D) = \frac{P(G,D)}{\sum_{G \in F} P(G,D)} \tag{3.16}$$

$$\hat{P}(f|D) = \sum_{G \in F} f(G)\hat{P}(G|D) \tag{3.17}$$

3.3.1.1 Approximation accuracy δ and the effect of k on δ

Let G_{MAP} denote a network with the best score (a MAP network). To measure the accuracy of the approximation, we consider the following quantity:

$$\delta = \sum_{G \in F'} \frac{P(G,D)}{P(G_{MAP},D)} \tag{3.18}$$

Proposition 1 *:*

$$0 \leq \frac{\hat{P}(G|D) - P(G|D)}{P(G|D)} \leq \delta \tag{3.19}$$

$$|\hat{P}(f|D) - P(f|D)| \leq (1 + \hat{P}(f|D))\delta \leq 2\delta \tag{3.20}$$

We can give an upper bound to δ. Let G_{min} be a network with the minimum score among the top-k best networks in F. A good upper bound to the number of possible DAGs is $C(n) = n!2^{\frac{1}{2}n(n-1)}$

Proposition 2 *:*

$$\delta < C(n)e^{-(score(G_{MAP}:D) - score(G_{min}:D))} \tag{3.21}$$

To achieve certain accuracy δ, we should compute the top-k best networks until

$$score(G_{MAP} : D) - score(G_{min} : D) \geq lnC(n) - ln\delta \tag{3.22}$$

For example, to achieve an accuracy of $\delta = 1\%$, the score difference should be atleast 50.90 for 10 variables, and atleast 178.64 for 20 variables.

CHAPTER 4. EXPERIMENTS AND RESULTS - SCORES AND POSTERIOR PROBABILITIES

In this Chapter, we find the top-k Bayesian networks for different datasets in the UCI Machine Learning Repository [Asuncion and Newman (2007)] for different k-values and make some observations about the Posterior probability and the BDe scores of the top-k networks.

4.1 Experiments Platform and Parameters

The algorithms have all been implemented in the C++ language and the experiments are run under Ubuntu on a laptop PC with 2.0GHz AMD Turion 64 X2 Mobile Technology TL-60 processor and 3.0GB memory.

The BDe score for $score_i(Pa_i : D)$ with a uniform structure prior $P(G)$ and equivalent sample size 1 was used. The Experiments were conducted on various datasets from the UCI Machine Learning Repository [Asuncion and Newman (2007)]: Iris, Balance, Nursery, Tic-Tac-Toe, Zoo, Letter Recognition and a synthetic dataset from a gold-standard 10-variable Bayesian network. All the datasets contain discretized variables (or are discretized) and have no missing values.

Table 4.1 Datasets used

Name	**n**	**m**
Iris	5	150
Balance	5	625
Nursery	9	12960
Tic-Tac-Toe	10	958
Zoo	17	101
Synthetic	10	1000
Letter Recognition	17	20000

4.2 Experimental Results

For each of the datasets in the Table 4.1, we run the algorithm to find the k-best networks for different k and then estimate the posterior probabilities of any hypotheses of interest $\hat{P}(h|D)$ as shown in Eqns 3.15.

We introduce a measure for the quality of the posterior estimation, the relative ratio of the posterior probability of the MAP network G_{map} to the posterior probability of the worst network G_{min} in the top-k networks. (i.e. the ratio of the score of the first top network to the score of the top-k^{th} network):

$$\lambda \equiv \frac{\hat{P}(G_{map}|D)}{\hat{P}(G_{min}|D)} \equiv \frac{P(G_{map}|D)}{P(G_{min}|D)} \tag{4.1}$$

It has been argued in [Madigan and Raftery (1994)] that we should make predictions using a set of the best models discarding those models which predict the data far less well even though the very many models with small posterior probabilities may contribute substantially to the sum. A cutoff value of $\lambda = 20$ is suggested in [Madigan and Raftery (1994)] by analogy with the 0.05 cutoff for P-values.

It can be noticed from Table 4.2 that the value of the quality measure λ increases as k value increases. Figure 4.1 and Figure 4.2 also show the value of $\hat{P}(G|D)$ for the top-k values plotted along the X-axis.

1. For the Iris case, the $top-32$ networks (with $\lambda = 103.122$) give a more reasonable approximation than the $top-10$ networks (with $\lambda = 2.446$). The MAP network has a posterior probability of 29.01% and the second one has a posterior probability of 19.32% as can be seen in Figure 4.1.

2. For the Balance case, it can be observed from Table 4.2 and Figure 4.1 that all the $top-5$ networks have the same $\hat{P}(G|D)$ with equal scores.

3. For the Nursery case, the $top-5$ networks itself give a reasonable approximation with a λ value of 360314.74. The top-2 networks have the same posterior probability of 0.469364.

4. For the Tic-tac-toe case, the top 150 networks seem to give a good approximation (λ = 277.4). In this case again, the $top - 76$ networks are equally probable and have the same BDe score. The top-76 networks are from two different equivalence classes but still have the same BDe score and the same posterior probability. Two of the $top - 76$ networks for tic-tac-toe are shown in Figure 4.2. This is a good example of a case where making inferences using just one of the networks (the MAP network) may lead to a poor approximation.

5. For the Zoo case, the graph in Figure 4.2 gives a good trend of the posteriors of the top-k networks with increase in k. Although there is only one network with the highest posterior probability, the other best networks have posteriors very close to that of the MAP network. This can be seen in the Figure 4.2. It can be observed that even for the top-100 networks, λ is only 1.52. Hence, the top 100 networks are also not enough to get a reliable estimation for Zoo.

6. For the Synthetic case, for k=100, λ is 27.158. The top-3 networks have the same maximum Posterior probability as shown in the Figure 4.2.

7. For the Letter Recognition case, k=50 gives a λ value of 52.32. Out of these 50 top networks, the top 18 networks are the most significantly contributing networks with the same posterior probability as seen in Figure 4.2.

We observed above that many networks may share the same posterior probability. This is because the BDe scoring criterion has the likelihood equivalence property, i.e., it assigns the same score to the Bayesian networks within the same independence equivalence class. However, it is also interesting to note that not all the networks that share the best posterior probability are in the same equivalence class. For example, in the Tic-tac-toe case, the top-76 networks share the same best posterior probability networks. This is shown in Figure 4.2. Hence, bayesian network structures with the same posterior probability may also be from different equivalent classes.

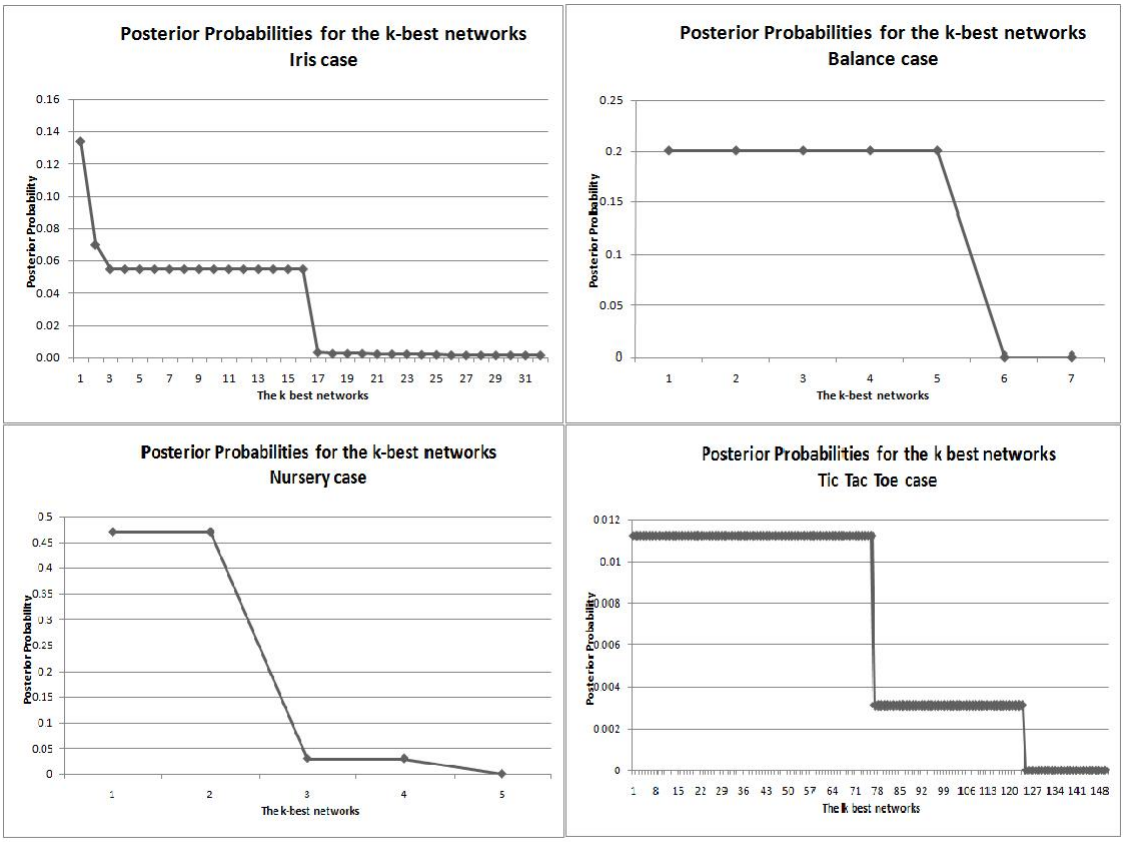

Figure 4.1 The Exact Posterior Probabilities of the k-best Networks.

Table 4.2 Experimental Results

Dataset Name	**n**	**m**
Iris	5	150

k-value	$Posterior Probability of first network$	$Posterior Probability of k^{th} network$	λ
k=1	1.00000000	1	1.000
k=10	0.20870300	0.08531640	2.446
k=32	0.13405900	0.001300000	103.122
k=100	0.12574600	0.000719971	174.654
k=325	0.11751700	0.000097834	1201.189
k=900	0.11652200	4.24E-006	27467.417

Dataset Name	**n**	**m**
Balance	5	625

k-value	$Posterior Probability of first network$	$Posterior Probability of k^{th} network$	λ
k=1	1.00000000	1	1.000
k=5	0.20000000	0.2	1.000
k=7	0.20000000	2.24E-013	8.92E+11

Dataset Name	**n**	**m**
Nursery	9	12960

k-value	$Posterior Probability of first network$	$Posterior Probability of k^{th} network$	λ
k=1	1.00000000	1	1
k=5	0.46936400	1.30E-006	3.60E+05

Dataset Name	**n**	**m**
Tic tac toe	10	958

k-value	$Posterior Probability of first network$	$Posterior Probability of k^{th} network$	λ
k=1	1.00000000	1	1
k=150	0.01119300	4.03E-005	277.4

Dataset Name	**n**	**m**
Zoo	17	101

k-value	$Posterior Probability of first network$	$Posterior Probability of k^{th} network$	λ
k=1	1.00000000	1	1.00
k=10	0.10521800	0.1	1.09
k=100	0.01291770	8.52E-003	1.52

Dataset Name	**n**	**m**
Synthetic dataset	10	1000

k-value	$Posterior Probability of first network$	$Posterior Probability of k^{th} network$	λ
k=1	1.00000000	1	1.000
k=100	0.07847200	0.00288948	27.158
k=1000	0.05282200	7.44E-005	709.629

Dataset Name	**n**	**m**
Letter Recognition	17	20000

k-value	$Posterior Probability of first network$	$Posterior Probability of k^{th} network$	λ
k=1	1.00000000	1.000	1
k=10	1.00000000	1.000	1
k=50	0.05296510	.001	52.32
k=100	0.05041360	.001	52.32

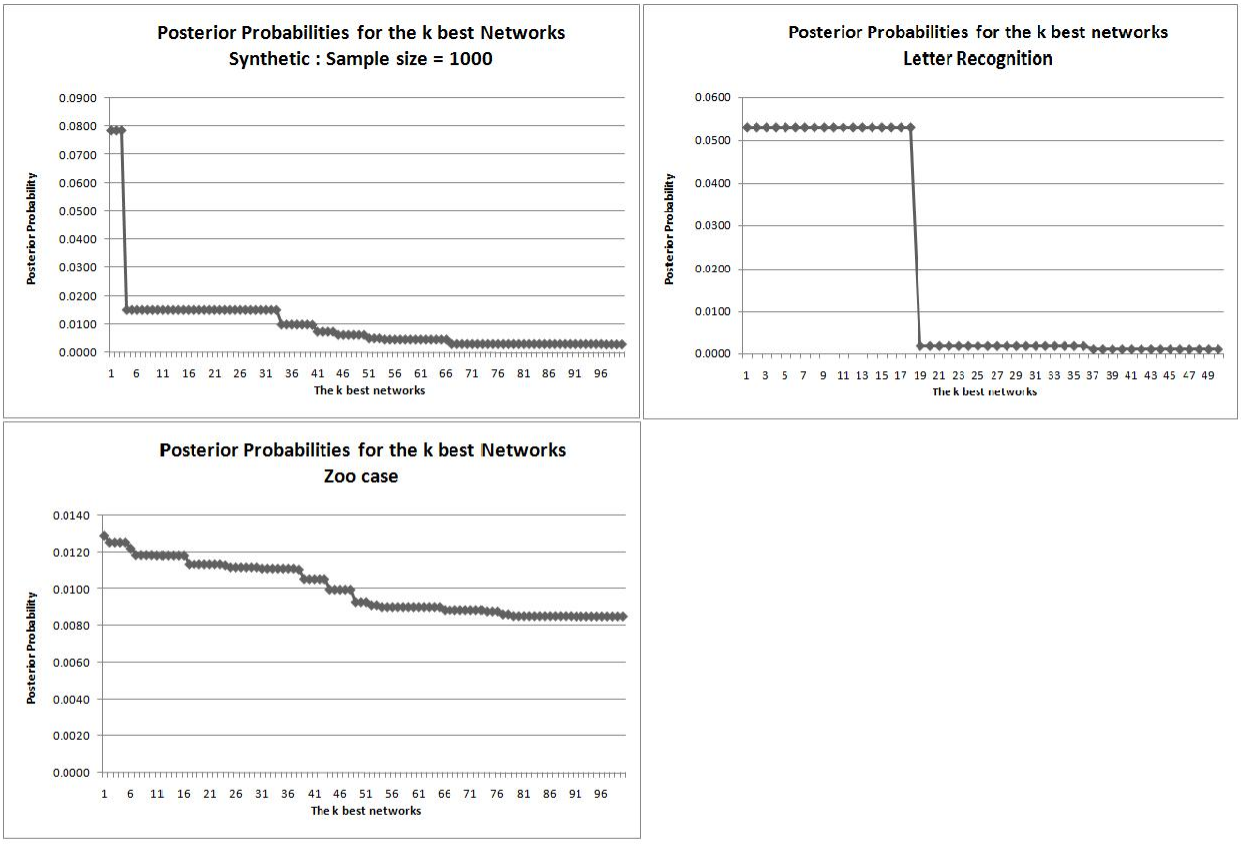

Figure 4.2 The Exact Posterior Probabilities of the k-best Networks (continued).

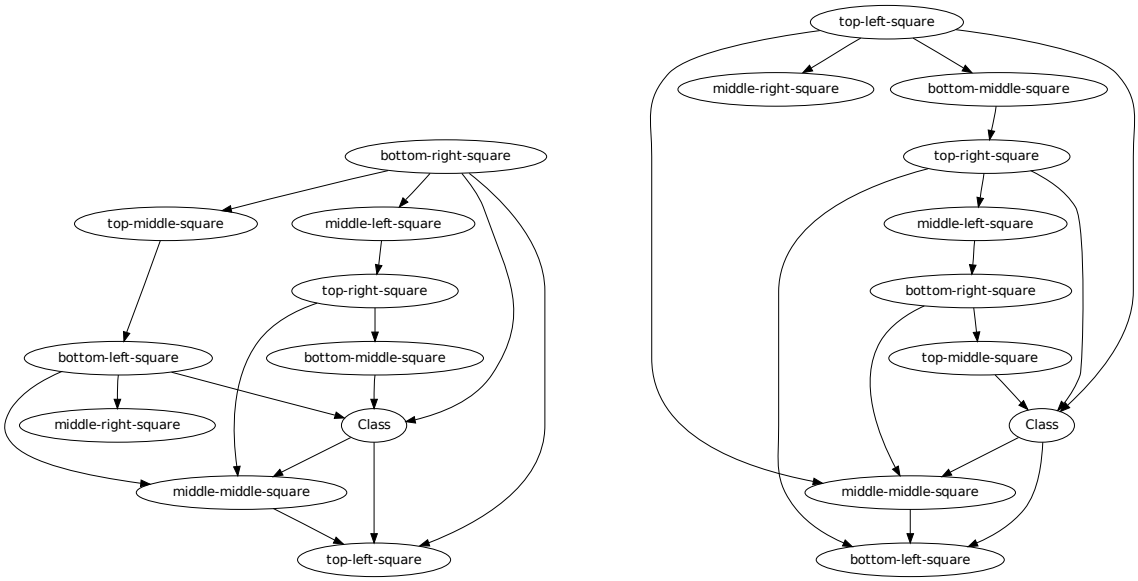

Figure 4.3 Two of the 76 top networks for the Tic-tac-toe dataset. These two networks share the same best posterior probability but have different skeletons

CHAPTER 5. EXPERIMENTS AND RESULTS - LEARNING THE HUMAN T-CELL SIGNALING CAUSAL MAP

In this chapter, we study the biological dataset CYTO and use the top-k model averaging method to learn the protein signaling network.

5.1 Introduction

In this section, we learn a protein signaling network from multicolor flow cytometry data that recorded the molecular activity of 11 proteins under various experimental conditions using the top-k Bayesian network structures. [Sachs *et al.* (2005)] modeled the protein signaling networks as causal Bayesian networks and inferred the network structure from the data using a Bayesian approach. More specifically, a random restart simulated annealing search is applied in the space of DAGs to find the networks with high posterior probabilities and a bootstrap method is used to find edges with high posteriors. The CYTO data was also analyzed by [Ellis and Wong (200)] using MCMC in the space of node orderings, and by [Eaton and Murphy (2007)] using a dynamic programming algorithm that computes the exact edge posterior probabilities under a special graph prior.

5.2 Revisiting the Bayesian Score

If the dataset D is drawn from a static distribution, closed form expressions for $P(D|G)$ have been derived [Cooper and Herskovits (1992), Heckerman *et al.* (1995)]. Assuming global and local parameter independence, and parameter modularity, the Bayesian score can be decomposed into the summation of local scores

$$score(G:D) = \sum_{i=1}^{n} score_i(V_i, Pa_i : D) \tag{5.1}$$

Further assuming Dirichlet parameter priors, we have

$$score_i(V_i, Pa_i : D) = aScore_i(A, D) \tag{5.2}$$

where

$$aScore_i(A, D) = ln \prod_{pa_i} \frac{\Gamma(\alpha_{pa_i})}{\Gamma(\alpha_{pa_i} + N_{pa_i})} \prod_{v_i} \frac{\Gamma(\alpha_{v_i;pa_i} + N_{v_i;pa_i})}{\Gamma(\alpha_{v_i;pa_i})} \tag{5.3}$$

where $\Gamma(.)$ is the Gamma function

N_{v_i,pa_i} is the number of cases in data set D for which V_i takes the value v_i and its parents Pa_i take the value pa_i

α_{v_i,pa_i} are Dirichlet hyper parameters

$A = \{\alpha_{v_i,pa_i} : v_i \in Dm(V_i), pa_i \in Dm(Pa_i)\}$

$\alpha_{pa_i} = \sum_{v_i} \alpha_{v_i,pa_i}$

$N_{pa_i} = \sum_{v_i} N_{v_i,pa_i}$

We use $\prod_{v_i}$ as a shorthand for $\prod_{v_i \in Dm(V_i)}$ and $\prod_{pa_i}$ for $\prod_{pa_i \in Dm(Pa_i)}$. Here, we will use the BDe score and assume the following hyperparameters

$$\alpha_{v_i,pa_i} = 1/(r_i q_i) \tag{5.4}$$

where r_i is the number of states of V_i and q_i is the number of states of Pa_i.

5.2.1 Bayesian score with interventions

The Bayesian score described above assumes that the dataset D is drawn from a static distribution. We can adapt the score to deal with the situation where we have a number of data sets $D^1, D^2, ...$, generated from the same causal structure but under different experimental conditions [Tian and Pearl (2001b)].

For example, assume that we have two data sets D^1 and D^2 , where D^1 is generated from the causal model $M =< G, \theta_G >$. Assume that D^2 is generated from M under an ideal

intervention on variable V_k that set V_k to a fixed value. Then the Bayesian score is given by [Cooper and Yoo (1999)]:

$$score_i(V_i, Pa_i : D^1, D^2) = \begin{cases} aScore_i(A, D^1 + D^2), i \neq k \\ aScore_i(A, D^1), i = k \end{cases} \tag{5.5}$$

If D^2 is generated from the same causal structure G but with different parameters θ'_G , and we have no knowledge about how the two sets of parameters θ_G and θ'_G differ, we may assume that they are independent and we use the following Bayesian score [Tian and Pearl (2001b)]:

$$score_i(V_i, Pa_i : D^1, D^2) = aScore_i(A, D^1) + aScore_i(A, D^2) \tag{5.6}$$

5.3 Data Preparation

The CYTO data consists of 9 data sets under different conditions as shown in Table 5.1. It has roughly 700 to 900 samples per experimental condition, corresponding to various interventions on the system of interest. The variables in each dataset are listed in Table 5.2.

Table 5.1 Datasets in CYTO

Dataset Name
cd3cd28
cd3cd28icam2
cd3cd28 + aktinhib
cd3cd28 + g0076
cd3cd28 + psitect
cd3cd28 + u0126
cd3cd28 + ly
pma
b2camp

Assume that the data set cd3cd28 is generated from the causal model $M =< G, \theta_G >$. Then we assume that each of the data sets cd3cd28+aktinhib, cd3cd28+g0076, cd3cd28+psitect, and cd3cd28+u0126 is generated from M under some ideal intervention. We will consider the data set cd3cd28+ly as generated from a general perturbation rather than an ideal intervention on

Table 5.2 Variables in CYTO datasets

Variable name
praf
pmek
plcg
PIP2
PIP3
p44/42 (erk)
pakts473
PKA
PKC
P38
pjnk

akt as the actual intervention is not directly on akt. We assume that each of the data sets cd3cd28icam2, cd3cd28+ly, pma, and b2camp is generated from the same causal structure G but with different parameters.

In summary, we use the following Bayesian score. For those variables on which no intervention is performed, i.e., for $V_i \in \{praf, plcg, PIP3, erk, P38, pjnk\}$

$$\begin{aligned} score_i(V_i, Pa_i : D) = aScore_i(A, D^{cd3cd28} + D^{u0126} + D^{g0076} + D^{psitect} + D^{aktinhib}) \\ + aScore_i(A, D^{icam2}) \\ + aScore_i(A, D^{ly}) \\ + aScore_i(A, D^{pma} + D^{b2camp}) \end{aligned} \tag{5.7}$$

where

$$\begin{aligned} M^{cd3cd28} &= D^{cd3cd28} + D^{u0126} + D^{g0076} + D^{psitect} + D^{aktinhib} \\ &= D^{cd3cd28+u0126+g0076+psitect+aktinhib} \end{aligned} \tag{5.8}$$

If a variable V_j is set by intervention in data set D^j , to compute the local score of V_j we will assume ideal intervention and simply drop the data set D^j from Equation 5.7.

Hence for the variables $V_i \notin \{praf, plcg, PIP3, erk, P38, pjnk\}$, we have

For $V_i = pmek$,

$$\begin{aligned} score_i(V_i, Pa_i) = aScore_i(A, D^{cd3cd28+g0076+psitect+aktinhib}) \\ +aScore_i(A, D^{icam2}) \\ +aScore_i(A, D^{ly}) \\ +aScore_i(A, D^{pma} + D^{b2camp}) \end{aligned} \tag{5.9}$$

For $V_i = PIP2$,

$$\begin{aligned} score_i(V_i, Pa_i) = aScore_i(A, D^{cd3cd28+u0126+g0076+aktinhib}) \\ +aScore_i(A, D^{icam2}) \\ +aScore_i(A, D^{ly}) \\ +aScore_i(A, D^{pma} + D^{b2camp}) \end{aligned} \tag{5.10}$$

For $V_i = pakts473$,

$$\begin{aligned} score_i(V_i, Pa_i) = aScore_i(A, D^{cd3cd28+u0126+g0076+psitect}) \\ +aScore_i(A, D^{icam2}) \\ +aScore_i(A, D^{ly}) \\ +aScore_i(A, D^{pma} + D^{b2camp}) \end{aligned} \tag{5.11}$$

For $V_i = PKA$,

$$\begin{aligned} score_i(V_i, Pa_i) = aScore_i(A, M^{cd3cd28}) \\ +aScore_i(A, D^{icam2}) \\ +aScore_i(A, D^{ly}) \\ +aScore_i(A, D^{pma}) \end{aligned} \tag{5.12}$$

For $V_i = PKC$,

$$\begin{aligned} score_i(V_i, Pa_i) = aScore_i(A, D^{cd3cd28+u0126+psitect+aktinhib}) \\ +aScore_i(A, D^{icam2}) \\ +aScore_i(A, D^{ly}) \\ +aScore_i(A, D^{b2camp}) \end{aligned} \tag{5.13}$$

After the local scores for each variable was calculated using the equations above, we run the top-k algorithm to find the top-100

5.4 Results and Discussion

Table 5.3 Posterior probability of top 10 networks based on calculation using k=500

Posterior probability
0.917081
0.053157
0.0297582
3.17847e-06
3.37237e-07
1.84235e-07
1.03138e-07
4.31118e-08
3.1489e-08
1.82521e-09

The k-value was chosen based on the λ value for different k. A cutoff of $\lambda = 20$ as discussed before was considered in choosing the k value. For k=500, the λ value is 1.15e+29. The posterior probability of the top-10 networks when we compute top-k with k=500 are shown in Table 5.3. The MAP network as shown in Figure 5.3 has a posterior probability of 91.70%. This network is unique and has the best score among all the top networks. The network obtained is very similar to the one obtained in [Sachs *et al.* (2005)]. Our method has 3 edge reversals and 4 new edges that were not present from the results in [Sachs *et al.* (2005)]. The biological significance of these discoveries could be an area of future study. By using the top-k model averaging, we were also able to find the posterior probabilities of the edges in the network as shown in Figure 5.1]. Most of the edges have a posterior probability of 1. The reversed edges could be due to the cyclic nature of causal relations in protein signaling networks.

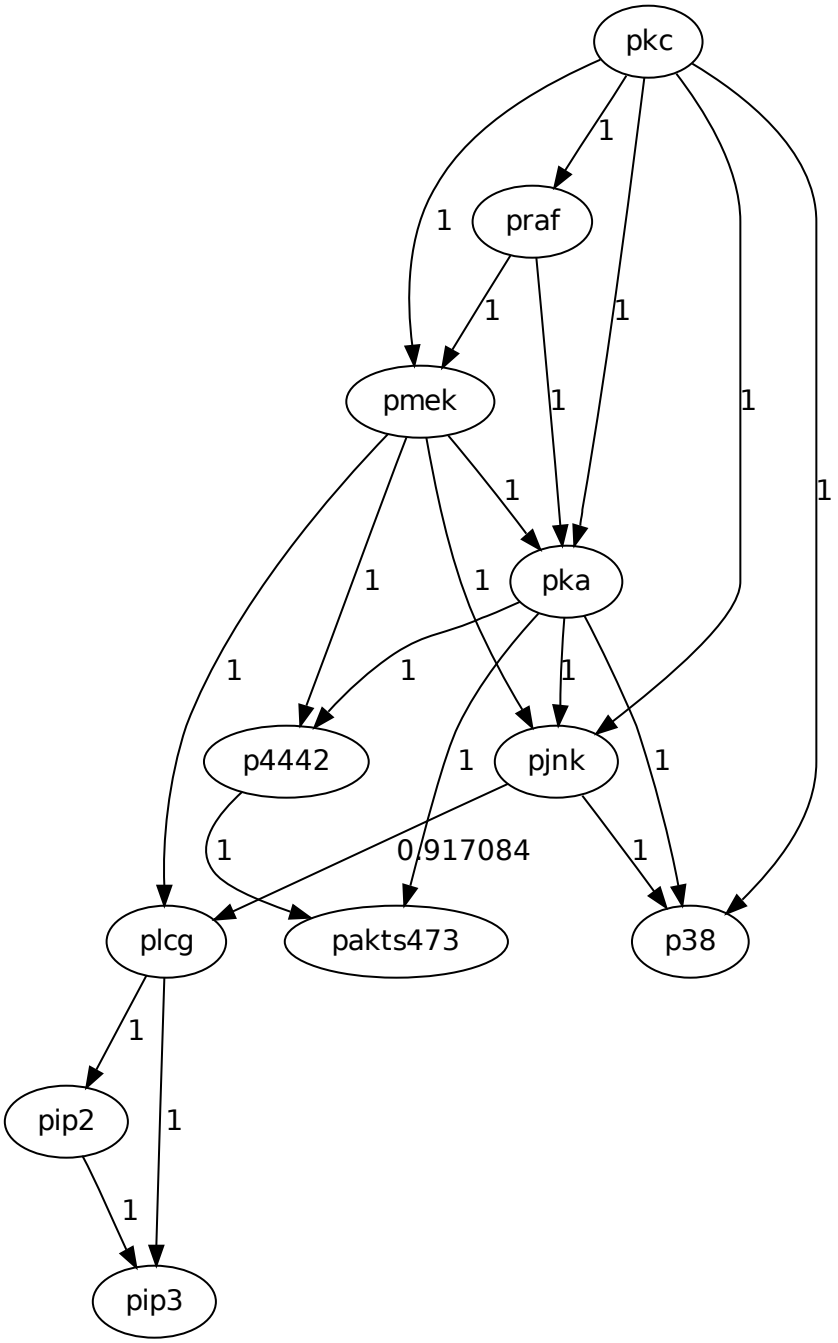

Figure 5.1 CYTO network obtained by averaging over top-500 Bayesian network structures. The edges are labeled with their edge posterior probabilities with only the most significant edges (with posterior > 0.70).

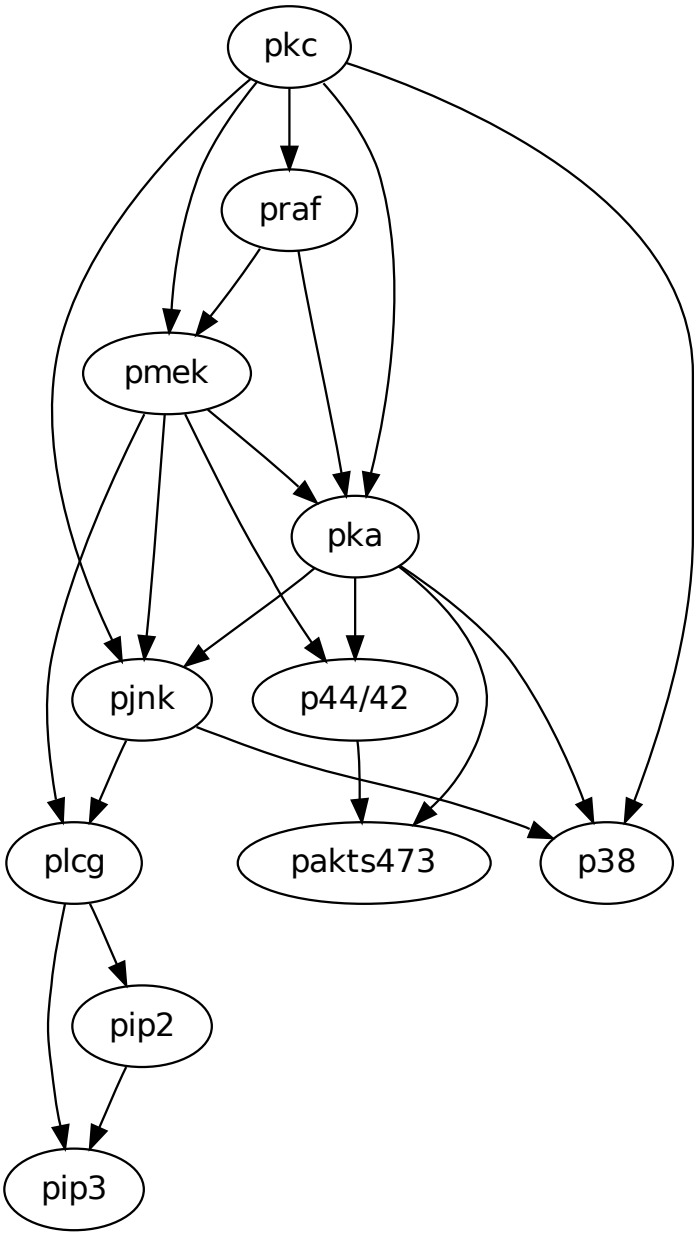

Figure 5.2 The MAP CYTO network obtained which has a posterior probability of 0.917081.

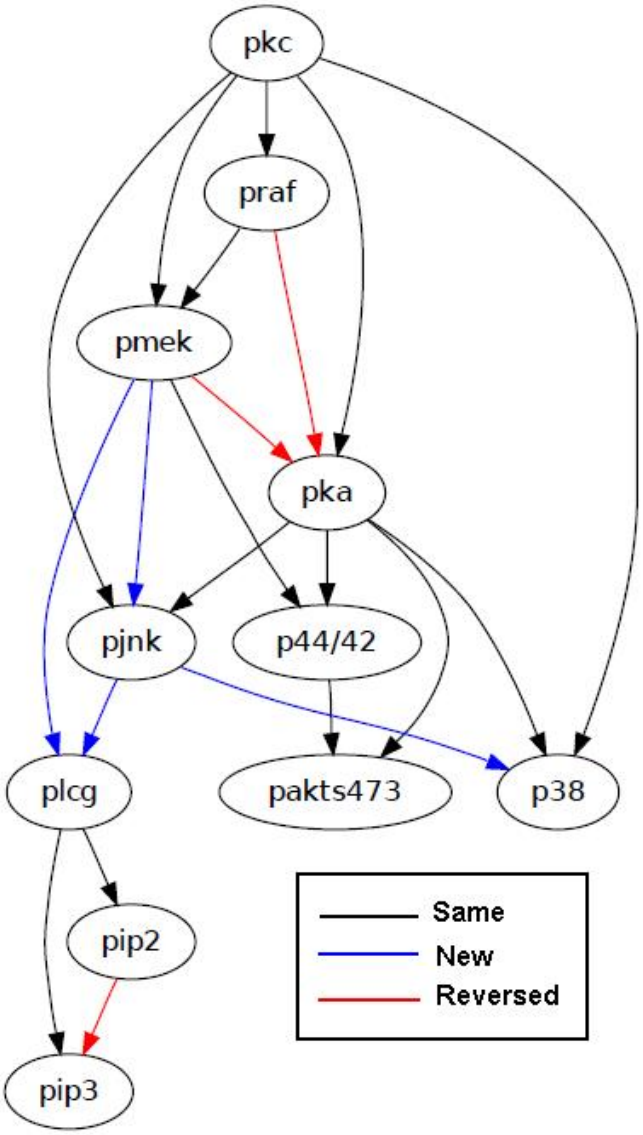

Figure 5.3 The network obtained by model averaging over top-500 networks compared with those in [Sachs *et al.* (2005)]. Edges marked 'Same' are same as in [Sachs *et al.* (2005)]. Edges marked 'Added' are found by our method but were not present in [Sachs *et al.* (2005)]. Edges marked 'Reversed' were reversed from [Sachs *et al.* (2005)]

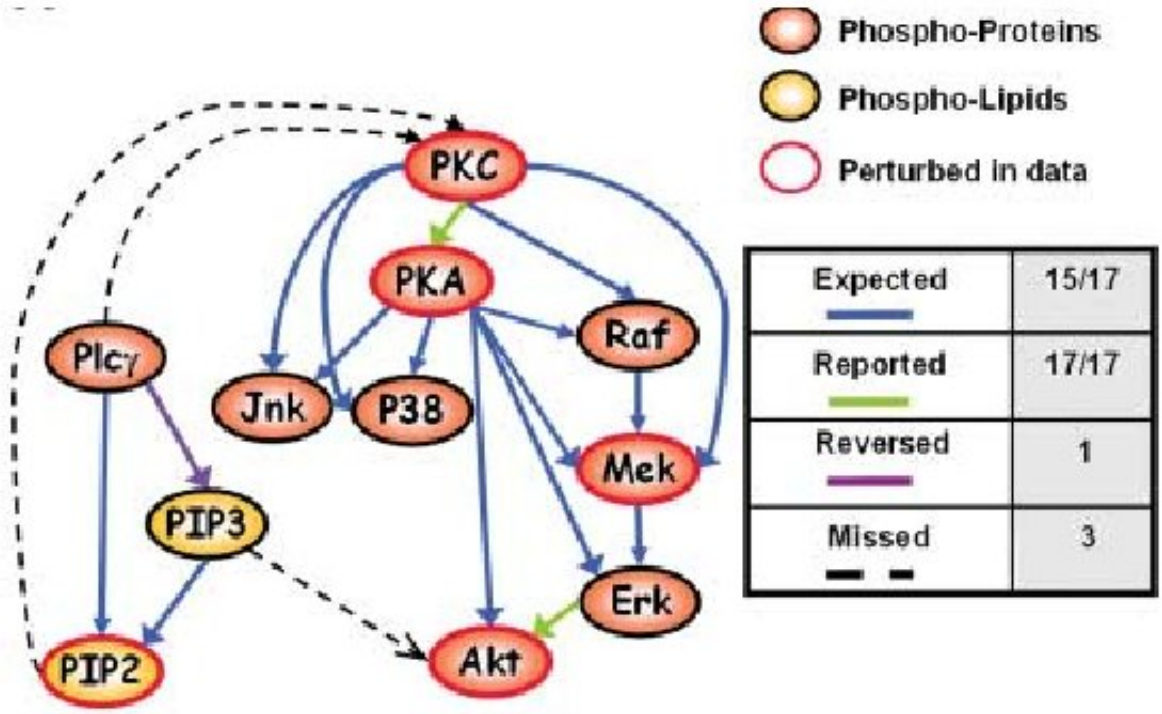

Figure 5.4 CYTO network obtained by [Sachs *et al.* (2005)] compared with those in the literature.

CHAPTER 6. EXPERIMENTS AND RESULTS - CLASSIFICATION PERFORMANCE OF TOP-k BAYESIAN NETWORKS

In this Chapter, we study the classification performance of the top-k Bayesian network structures.

6.1 Experiments Idea and Datasets

The idea behind using the top-k networks for prediction tasks is an intuition that a better estimation of the posterior $P(G|D)$ will also perform better in terms of prediction. Hence, if we are able to do a model averaging over all the best-k networks which may have a significant contribution towards the posterior, we hope to approximate the value $P(G|D)$ more closely.

The Experiments were conducted on various datasets from the UCI Machine Learning Repository [Asuncion and Newman (2007)]: Balance, Nursery, Tic-Tac-Toe, Zoo and a synthetic dataset from a gold-standard 10-variable Bayesian network as shown in Table 6.1. All the datasets contain discretized variables (or are discretized) and have no missing values.

The Figure 6.1 shows the percentage split of the dataset into training and test sets.

We use a simple metric, Match Ratio to evaluate the classification accuracy:

$$Matchratio = \frac{Numberofcorrectpredictions}{Totalnumberofpredictions} \tag{6.1}$$

6.2 Results and Discussion

The classification results are shown in Figure 6.1. The k-value is chosen based on a cutoff of the λ value to be 20. We see that for the datasets Balance and Nursery we see a small improvement in the classification match ratio. However, for the tic-tac-toe dataset we see a

Table 6.1 Datasets used

Name	**n**	**m**
Balance	5	625
Nursery	9	12960
Tic-Tac-Toe	10	958
Zoo	17	101
Synthetic	10	1000

significant improvement in the classification accuracy. This might be because of the observation made in Figure 4.2 that there are multiple networks sharing the same best posterior probability that may have different skeletons and are from multiple equivalence classes. When we average over all the top-k networks, this may lead to a better fitting of all the models during prediction. We also observed that the Zoo and the synthetic dataset did not show any improvement in prediction accuracy because of using the top-k models over the best MAP network alone. For the zoo dataset, this could be because even the top-100 networks had a λ value of only 1.52.

Dataset Name	**n**	**N**	
Balance	5	625	
	N_train	**N_test**	**Match ratio on test set**
K=1	438	187	86.0963%
K=7	438	187	88.23%

Dataset Name	**n**	**N**	
Nursery	9	12960	
	N_train	**N_test**	**Match ratio on test set**
K=1	11960	1000	93.60%
K=5	11960	1000	93.80%

Dataset Name	**n**	**N**	
Tic tac toe	10	958	
	N_train	**N_test**	**Match ratio on test set**
K=1	671	287	86.41%
K=150	671	287	96.17%

Dataset Name	**n**	**N**	
Zoo	17	101	
	N_train	**N_test**	**Match ratio on test set**
K=1	81	20	95.00%
K=100	81	20	95.00%

Dataset Name	**n**	**N**	
Synthetic dataset	10	1000	
	N_train	**N_test**	**Match ratio on test set**
K=1	666	334	55.68%
K=1000	666	334	55.68%

Figure 6.1 Classification Results of top-k Bayesian model averaging. The k-value is chosen based on a cutoff value for λ to be 20

CHAPTER 7. SUMMARY AND DISCUSSION

We develop an algorithm for finding the k-best Bayesian network structures. We make some observations about the posterior probability $P(G|D)$ and its value for different k-values. As we studied, there may be many structures which have the same best posterior probability and making conclusions based on just one MAP network will not always give a good approximation of the underlying network. A value of $\lambda = 20$ was considered a reasonable threshold to consider the top-k approximation to be a close one.

As the results show, many networks in the set of best networks may be in the same equivalent class. The algorithm searches for the top-k networks irrespective of their inclusion in the same equivalence class or not. Extension of the dynamic programming idea to search for best networks in the space of equivalence classes rather than in the space of all networks could be an area of future study.

We also show the results of the top-k model averaging on the CYTO protein signaling network. We observed that the network found by our method is very close to the results from [Sachs *et al.* (2005)]. We were also able to find the posterior probability of each edge in the protein signaling network. However, using techniques to study cyclic network structures may give a better inference about the protein signaling networks owing to their highly feedbacking network structure.

We also study the classification performance of the top-k model averaging technique and observe that the averaging over the top-k networks is more useful in some cases than others. From the results, it seems like networks with many best networks which may belong to multiple equivalence classes may benefit from the top-k averaging over the single MAP network's prediction owing to the diversity in the network structures that fit the data.

BIBLIOGRAPHY

J. Pearl (2000). Causality: Models, Reasoning, and Inference. *Cambridge University Press, NY, 2000.*

P. Spirtes, C. Glymour, and R. Scheines (2001). Causation, Prediction, and Search (2nd Edition). *MIT Press, Cambridge, MA, 2001.*

D. M. Chickering. (1996). Learning Bayesian networks is NP-complete. *In D. Fisher and H. J. Lenz, editors, Learning from Data: Artificial Intelligence and Statistics V. Springer Verlag, 1996.*

Günther Witzany. (2010). Biocommunication and Natural Genome Editing. *Springer.*

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A. Lauffenburger, Garry P. Nolan. (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science, Vol.308, No.5721, pp.523-529, 2005.*

Mitchell Koch, Bradley M. Broom,and Devika Subramanian. (2009). Learning robust cell signaling models from high throughput proteomic data. *Int. J. Bioinformatics Research and Applications, Vol.5, No.3, pp.241-253, 2009.*

D. Heckerman, C. Meek, and G. Cooper. (1999). A Bayesian approach to causal discovery. *In Glymour C. and Cooper G.F., editors, Computation, Causation, and Discovery, Menlo Park, CA, 1999. AAAI Press and MIT Press.*

D. Madigan and J. York. Bayesian graphical models for discrete data. *International Statistical Review, 63:215-232, 1995.*

Nir Friedman and Daphne Koller. Being Bayesian about network structure: A bayesian approach to structure discovery in bayesian networks. *Machine Learning, 50(1-2):95-125, 2003.*

M. Koivisto and K. Sood. Exact Bayesian structure discovery in Bayesian networks. *Journal of Machine Learning Research, 5:549-573, 2004.*

M. Koivisto. Advances in exact Bayesian structure discovery in Bayesian networks. *In Proceedings of the Conference on Uncertainity in Artificial Intelligence (UAI), 2006.*

J. Tian and R. He. Computing posterior probabilities of structural features in Bayesian networks. *In Proceedings of the Conference on Uncertainity in Artificial Intelligence (UAI), 2009.*

D. Madigan and A. Raftery. Model selection and accounting for model uncertainity in graphical models using occam's window. *Journal of American Statistical Association, 89:1535-1546, 1994.*

D. Heckerman, C. Meek, and G. Cooper. A Bayesian approach to causal discovery. *Technical Report MSR-TR-97-05, Microsoft Research, 1997.*

Ajit P. Singh and Andrew W. Moore. Finding optimal Bayesian networks by dynamic programming. *Technical report, Carnegie Mellon University, School of Computer Science, 2005.*

T. Silander and P. Myllymaki. A simple approach for finding the globally optimal Bayesian network structure. *In Proceedings of the Conference on Uncertainity in Artificial Intelligence (UAI), 2006.*

G. F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning, 9:309-347, 1992.*

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

A. Asuncion and D. J. Newman. *UCI machine learning repository, 2007.*

Nir Friedman, Moises Goldszmidt, and Abraham Wyner. Data analysis with bayesian networks: A bootstrap approach. *In Proceedings of the Fifteenth Annual Conference on Uncertainity in Artificial Intelligence (UAI-99), pages 196-200, San Fransisco, CA, 1999.*

D. Eaton and K. Murphy. Bayesian structure learning using dynamic programming and MCMC. *In Proceedings of the Conference on Uncertainity in Artificial Intelligence, 2007.*

Russell, Stuart J , and Peter Norvig. Artificial Intelligence: A Modern Approach. *Prentice Hall series in artificial intelligence. Upper Saddle River, N.J: Prentice Hall/Pearson Education, 2003. Print.*

Richard E. Neapolitan. Learning Bayesian Networks *Prentice Hall, April 2003.*

B. Ellis and W. H. Wong. Learning causal Bayesian network structures from experimental data. *Journal of American Statistical Association, 103:778-789, 2008.*

D. Eaton and K. Murphy. Exact Bayesian structure learning from uncertain interventions. *In AI/Statistics, 2007.*

J. Tian and J. Pearl. Causal Discovery from changes: a Bayesian approach. *Technical Report R-285, Department of Computer Science, University of California, Los Angeles, 2001b*

G. F. Cooper and C. Yoo. Causal discovery from a mixture of experimental and observational data. *In Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI99), pages 116125, San Francisco, CA, 1999. Morgan Kaufmann Publishers.*