Master in Artificial Intelligence (UPC-URV-UB)

Master of Science Thesis

Bayesian Gaussian network classifiers for mass spectra classification

Víctor Manuel Bellón Molina

Advisor: Jesús Cerquides Bueno

18 January 2013

Contents

1 Introduction **5**
- 1.1 Proteomics . . . 5
- 1.2 Probabilistic classification . . . 6
- 1.3 Structure of the thesis . . . 7
- 1.4 Contributions . . . 7

2 Probabilistic Graphical models **8**
- 2.1 Graph theory . . . 8
- 2.2 Bayesian and Markov networks . . . 11
  - 2.2.1 Bayesian networks . . . 11
  - 2.2.2 Markov networks . . . 12
  - 2.2.3 Equivalences between Bayesian and Markov Networks . 12
  - 2.2.4 Markov distributions over decomposable graphs . . . 13
- 2.3 Conditional Gaussian networks . . . 14
- 2.4 Gaussian Join tree classifiers . . . 15

3 Learning PGM **17**
- 3.1 Maximum likelihood . . . 17
- 3.2 Bayesian model averaging . . . 18
  - 3.2.1 Hyper Markov distributions over decomposable graphs 19

4 Learning CGN by ML **20**
- 4.1 The multinomial distribution . . . 20
  - 4.1.1 Parameter learning . . . 20
- 4.2 The normal distribution . . . 21
  - 4.2.1 Parameter learning . . . 21
  - 4.2.2 Condtional Gaussian distribution . . . 21
- 4.3 Learning conditional Gaussian Networks . . . 22

5 Learning GJTc by BMA **23**
- 5.1 Dirichlet distribution . . . 23

5.1.1 Conjugancy 24
5.1.2 Predictive posterior distribution 24
5.2 Hyperinverse Wishart distribution 24
5.3 Hyper normal inverse Wishart distribution 25
5.3.1 Conjugacy 26
5.3.2 Predictive distribution 26
5.4 Gaussian Join Tree Classifiers 26
5.5 Predicting 28

6 Empirical Comparison 29
6.1 Spectogram datasets 29
6.1.1 Ovarian cancer dataset 29
6.1.2 Pancreatic cancer dataset 29
6.2 Dependency structures 30
6.3 k-BOX structure 30
6.3.1 k-BAND structure 31
6.4 The prior 31
6.5 The experiment 32
6.5.1 Measures 32
6.6 Results 33
6.6.1 Ovarian data 33
6.6.2 Pancreatic data 33

7 Conclusions and Future Work 36

A Distributions 39
A.1 Multivariate t-distributions 39
A.1.1 Marginals 39
A.2 Normal Inverse Wishart distributions 39
A.2.1 Conjugacy 40
A.2.2 Predictive distribution 40
A.2.3 Marginals 41

List of Figures

2.1 Example of a directed graph. 9
2.2 Example of an undirected graph. 9
2.3 Naive Bayes structure, with four predictive variables. 15

6.1 Structure of the graph for k-BOX model 30
6.2 Structure of the covariance matrix for k-BOX model 30
6.3 Structure of the graph for k-BAND model 31
6.4 Structure of the covariance matrix for k-BAND model 31
6.5 Prediction of ovarian cancer. Accuracy versus number of parameters in model. 34
6.6 Prediction of ovarian cancer. CLL versus number of parameters in model. 34
6.7 Prediction of pancreatic cancer. Accuracy versus number of parameters in model. 35
6.8 Prediction of pancreatic cancer. CLL versus number of parameters in model. 35

Chapter 1

Introduction

The early diagnosis of diseases in patients is a key objective of biomedical science and one of the most important factors in the treatment of diseases such as cancer. The early detection of cancer can make the difference between a successful treatment and the dead of the patient.

Ovarian cancer is diagnosed at late clinical stage in more than 80% of patients and the 5-year survival rate is around 35% of population, while in early diagnosed patients it exceeds 90% [III et al.2002]. The aim of this work is to present techniques for the early detection of ovarian cancers based in probabilistic analysis of proteomic spectra.

1.1 Proteomics

A proteome is the entire protein complement of a cell, tissue, or organism in a particular state [Elo and Schwikowski2011]. With the advances of the later years the focus of biomedical research has moved from protein identification towards the analysis of proteome. Among all possible techniques, mass spectrometry has become popular due to its ability to deal with a wide range of different proteins in complex proteomes.

Gene expression is a complex process where information from a gene is transcribed, by synthesis, in a gene product. This products are often proteins, but they also could be functional RNA. One important component of gene expression is protein expression. Protein expression starts after DNA is transcribed into messenger RNA (mRNA). The process consist in the translation of mRNA to polypeptide chains, which are ultimately folded into proteins.

The sequencing technologies of mRNA are becoming cheaper and increasing their availability. However the analysis of mRNA does not give any information about whether a protein is being expressed or not. Even though the mRNA formation is the first step of the protein synthesis it does not give all the information about which proteins are going to be expressed because different proteins can be generated from a single gene.

Proteomic study is becoming an important tool in biomedical research. A key objective of the proteomic analysis is the early detection of diseases, expected to lead to a longer survival rate. Unfortunately proteomic data consists in a large set of data which requires the use of classification tools.

1.2 Probabilistic classification

Supervised classification is a basic task in data analysis and pattern recognition. It requires the construction of a classifier, that is, a function that assigns a class label to instances described by a set of variables. There are numerous classifier paradigms, among which the ones based on probabilistic graphical models (PGMs) [Lauritzen1996], are very effective and well-known in domains with uncertainty.

PGMs can be divided in three main groups, those with discrete variables, those with continuous variables and those with both types of variables which are called mixed models. Probabilistic classifiers have a discrete variable corresponding to the class. Therefore there will be only two family models when we talk about classification, discrete models, if all the variables are discrete, and mixed models, when one or more variables are continuous.

Also, a widely used assumption is that data follows a multidimensional Gaussian distribution[Geiger and Heckerman1994]. This is adapted for classification problems by assuming that data follows a multidimensional Gaussian distribution that is different for each class, encoding the resulting distribution as a Conditional Gaussian Network (CGN)[Bøttcher2004].

In [Larrañaga et al.2006], Larrañaga, Pérez and Inza introduce and evaluate classifiers based on CGNs with a more detailed description in [Pérez2010]. They analyze different methods to identify a Bayesian network structure and a set of parameters such that the resultant CGN performs well in the classification task. In [Pérez2010] the authors propose to estimate the parameters directly from the sample mean and sample covariance matrix in the data, that is, using maximum likelihood. It is well known [Buntine1996] that following this strategy can lead to model overfitting when data is scarce.

In bioinformatics, models are sought in domains where the number of data items is small and the number of variables is large, such as classification of mass spectrograms or microarrays. To try to avoid overfitting, we introduce classifiers based on Gaussian Join Trees (GJT) that instead of estimating by maximum likelihood perform exact Bayesian averaging over the parameters.

1.3 Structure of the thesis

We start by shortly reviewing in the theory of Probabilistic Graphical Models and conditional Gaussian networks and introducing Gaussian Join tree classifiers in chapter 2. Next we get into the learning problem. In chapter 3 we review the Maximum likelihood and the bayesian model averaging methods. Also we review the theory oh Hyper Markov distribution over decomposable graphs.

Chapter 4 is used to explain the learning process over conditional Gaussian Networks using maximum likelihood. Following, in chapter 5 we explain the process to learn Gaussian join tree classifiers using Bayesian model averaging and also introduce the Hyper Normal Inverse Wishart distribution.

In chapter 6 we perform a comparison between models learned using maximum likelihood and Bayesian model Averaging. The comparison is based in classifying mass spectrometry data from cancer patients. Finally in chapter 7 we give the final conclusions of the work.

1.4 Contributions

The contributions of the work are:

1. The proposal of GJT classifiers including a proof that the $\mathcal{HNIW}$ law is strong hyper Markov,
2. A preliminary empirical comparison with classifiers adjusting parameters by maximum likelihood for the task of mass spectra classification and,
3. An open source implementation of the algorithms, made available for easy reproducibility of the results.

Chapter 2

Probabilistic Graphical models

A probabilistic graphical model is a representation of a join probability distribution over a set of variables. It has two main parts, one is the structure, i.e., the set of conditional dependences between the which is represented by a graph, and the other is the conditional distributions over the variables in which the join distribution factorises.

This chapter introduces basic concepts in probabilistic graphical models. We start by reviewing some necessary concepts on graph theory. Next in section 2.2 we take a look on Bayesian networks and Markov networks and review important results on the equivalence of Bayesian and Markov networks.In the same section, we review Markov distributions over decomposable graphs theory. Following, we review the concept of conditional Gaussian networks introduced by [Bøttcher2004] and see how apply them to classification. Finally we introduce Gaussian Join Tree classifiers, an adaptation of conditional Gaussian networks to Markov networks which take advantage of the Markov distributions theory.

2.1 Graph theory

Graph theory is an accepted formalism for representation of probabilistic graphical models. In this section we introduce the basic definitions and notations that will be used. Most of the definitions of this section have been adapted from [Koller and Friedman2009].

We start by introducing a general definition of graph.

Definition 1. A **graph** is a pair $G = (X, E)$ where X is a non-empty finite set of vertices and E is a set of pairs of nodes of X called edges. We call an edge $e = (x_i, x_j) \in E$ for $x_i, x_j \in X$ a **directed edge** if $(x_i, x_j) \neq (x_j, x_i)$, otherwise we said that e is an **undirected edge**. If all the edges of a graph

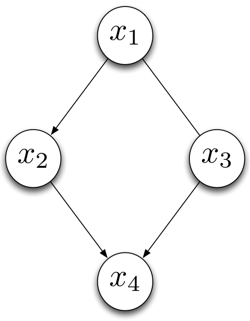

Figure 2.1: Example of a directed graph.

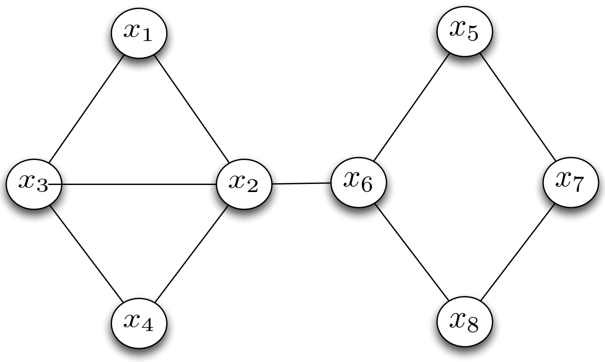

Figure 2.2: Example of an undirected graph.

G are directed, we call G a **directed graph**, on the other hand if all of the edges of graph are undirected we call it an **undirected graph**. A graph is a **complete graph** if for any ordered pair, $x_i, x_j \in X$ there is an edge $e = (x_i, x_j) \in E$.

The general agreement is to represent a graph using arrows and lines that connect nodes represented by different elements, usually a circle or a point. Arrows represents direct edges and lines undirected edges. An example of a directed graph is shown in figure 2.1, and an example of an undirected graph is shown in figure 2.1.

An important concept when we work with directed graphs is the one of parent and child.

Definition 2. Given a graph $G = (X, E)$ and an edge $e = (x_i, x_j)$ we say that x_i and x_j are **adjacent** nodes. If e is directed we say that x_i is a **parent** of x_j and x_j is a **child** of x_i. If e is undirected and x_i and x_j are connected by an edge then x_i and x_j are **neighbours**.

In general the parent is represented as the node at the tail of the arrow, and the child is the one at the head. For example, in figure 2.1 x_1 is the parent of x_2 and x_3 which are children of x_1. In figure 2.1 we have that the neighbours of x_3 are x_1, x_2 and x_4 Now we have defined what a graph is, and its different components, but sometimes we just need to work with a group of nodes of the graph.

Definition 3. Let $G = (X, E)$ be a graph and $\bar{X} \subset X$. We say that $H = (\bar{X}, \bar{E})$ is the induced **subgraph** by X of G if any edge $e = (x_i, x_j)$ such as $x_i, x_j \in \bar{X}$ is in $\bar{E}$, and there is not any other edge in $\bar{E}$. If H is complete it is called **clique**, and it is said to be a **maximal clique** if for any $Y \supset \bar{X}$, the induced subgraph by Y is not a clique.

For example in figure 2.1 Usually we will abuse notation and use clique to refer to the maximal cliques and speak in general of subgraphs instead of using the term induced subgraphs. Finally we need to talk about paths.

Definition 4. We say that a sequence $x_1, \dots, x_k$ forms a **path** in the graph $G = (X, E)$ if for every $i = 1, \dots, k-1$ we have that $(x_i, x_{i+1}) \in E$. If at least one edge in the path is directed, then we say that it is a **directed path**. A **cycle** in G is a directed path $x_1, \dots, x_k$ where $x_1 = x_k$. A graph is **acyclic** if it contains no cycles.

For example in figure 2.1 the sequence x_1, x_2, x_4 is a path, but x_4, x_3, x_1 is not a path, because its not correctly oriented. Also noticed that the graph is acyclic since there is no path from one variable to itself. A more general notion of path is the one of trail, in which we don't care about the direction of the edges.

Definition 5. A sequence $x_1, \dots, x_k$ form a **trail** in the graph $G = (X, E)$ if for every $i = 1, \dots, k-1$ we have that (x_i, x_{i+1}) is in E or (x_{i+1}, x_i) in in E. A **loop** in G is a trail $x_1, \dots, x_k$ where $x_1 = x_k$. A **chord** in a loop is an edge connecting two non consecutive nodes.

Now we define the concept of moral graph that will take importance later.

Definition 6. A directed graph is said to be a **moral graph** if there is an edge between the parents of each node.

If we look at the graph in figure 2.1 we can notice that it is not moral, since there is no edge between x_2 and x_3.

Definition 7. An undirected graph is said to be **chordal** if any loop of the graph $x_1, \dots, x_k, x_1$ for $k \geq 4$ has a chord. A graph G is said to be **chordal** if its underlying undirected graph is chordal.

Taking the subgraph induced by the nodes x_1, x_2, x_3, x_4 from the graph in figure 2.1 we get an example of a chordal graph, but if we take the subgraph induced by nodes x_5, x_6, x_7, x_8 we get a non chordal graph since we have a cicle of length 4 with no chords.

Definition 8. A pair (A, B) of subsets of nodes X is said to form **a decomposition** of $G = (X, E)$ if $X = A \cup B$ and A is complete and it separates A from B.

We can see that in figure 2.1, if we take the subgraph formed by x_1, x_2, x_3 and x_4 we can see it is decomposable taking x_1, x_2, x_3 and x_2, x_3, x_4 as the parts to be decomposed.

2.2 Bayesian and Markov networks

As we have seen in the previous section (2.1) there are two main families of graphs: directed graphs and undirected graphs. Bayesian network models use directed graphs as structure of the model, and Markov networks use undirected graphs. In this section we will see the main difference between the two models and also explain when they are equivalents.

2.2.1 Bayesian networks

Following we give a definition Bayesian networks and take a look on its factorization.

Definition 9. A Bayesian network structure G is a directed acyclic graph whose nodes represent random variables $X_1, \ldots, X_n$. G encodes the following set of independence assumptions, $I_l(G)$, also called local independences:

- For each variable X_i, X_i is independent of their non descendant variables given its parents Pa_i.

In general if a distribution P is associated with a Bayesian Network structure and the local independences encoded in G holds for P we said that G is an I-map of P. Also it allows to factorize the distribution using the following formula.

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(x_i | Pa_i)$$

When we define a Bayesian network we can think about causality as a heuristic way to decide what edge should be there. An edge between two nodes means a direct relation of causality between the two nodes: The parent causes the children. This way of thinking is helpful when we are creating a model. A good example will be a medical network where we have some illnesses and some symptoms. The symptoms are a direct consequence of the illnesses, therefore we will have direct edges from the illnesses to their respective symptoms. But when we are putting directed edges between nodes what we are assuming is a set of conditional independences.

2.2.2 Markov networks

We have already talked about Bayesian networks, now we will talk about Markov networks. We start by giving the following definition.

Definition 10. A Markov network structure G is an undirected graph whose nodes represent random variables $X = X_1, \dots, X_n$. We define the Markov blanket of X_i, $MB(X_i)$ in G to be the neighbours of X_i in G. Then G encodes the following set of local independence assumptions:

- For each variable X_i, X_i is independent of $X \setminus (X_i \cup MB(X_i)$ given its Markov blanket.

We have seen which kind of dependency we are encoding when we use an undirected edge. We can use as heuristic the following idea: We put an undirected edge between to nodes whenever both nodes are related, but not necessarily with a cause-effect relation. Imagine the following toy example: we have a network with two variables, one is the probability that my garden is wet, and the other the probability that my neighbour's garden is wet. There is no causality relation between the event of both variables but if my garden is wet, it is likely that the garden of my neighbour is also wet, and vice versa. Again by using a Markov Network structure we state a set of local independences.

2.2.3 Equivalences between Bayesian and Markov Networks

Until now we have seen that Markov Networks and Bayesian Networks code different local independences, i.e., they encode different models. The following two results tell us when a Markov Network and a Bayesian Network encode the same set of independences.

Definition 11. The moral graph $M(G)$ of a Bayesian network structure G over X is the undirected graph over X that contains an undirected edge between two nodes if:

- there is a directed edge between them, or
- they are parents of the same node.

Proposition 1. *If the directed graph G is moral, then its moralized graph $M(G)$ encode the same conditional independences than G.*

Proposition 2. *Let H be a chordal Markov network. Then there is a Bayesian network G such that the independences encoded in G and H are the same.*

Summing up, the only way that the independences in a graph can be represented exactly in both types of models is if and only if the graph is chordal. Therefore chordal graphs are the intersection between Markov and Bayesian Networks.

2.2.4 Markov distributions over decomposable graphs

We have seen that an important class of models is the class of model with a chordal graph as structure. A graph is in fact chordal if and only if it is decomposable. When we are working with a decomposable graph the join probability distribution can be factorized in terms of the cliques of the graph. This theory is introduced in [Dawid and Lauritzen1993]. Here we review the main results.

In the following, let $\mathcal{G} = (V, E)$ be an undirected decomposable graph over a set of random variables. A graph is said to be decomposable if it can be decomposed into its cliques.

Definition 12. A distribution P on V is called Markov over $\mathcal{G}$ if for any decomposition (A, B) of $\mathcal{G}$

$$A \perp\!\!\!\perp B | A \cap B.$$

Where $A \perp\!\!\!\perp B$ means that A and B are independent. In other words, a distribution is Markov if given a descomposition of a graph, the probability of the two components is independent given their intersection.

In general a graphical model $M(\mathcal{G})$ is a family of probability distributions which are Markov over $\mathcal{G}$.

Definition 13. We say that distributions Q over A and R over B are consistent if both yield the same distribution over $A \cap B$.

The next theorem tells us that given a set of marginal probability distributions over the cliques of a graph that are consistent between them, we can assess the unique joint probability distribution having those marginals. Let $\mathcal{C}$ be the set of cliques of $\mathcal{G}$, and $\mathcal{S}$ the corresponding collection of separators (including possible repetitions) (see [Dawid and Lauritzen1993, Cowell et al.1999] for details).

Theorem 1. *Given a pairwise consistent collection of distributions $\{Q_C : C \in \mathcal{C}\}$, Q_C being a distribution over C, the unique Markov distribution over $\mathcal{G}$ having $\{Q_C\}$ as its marginals is*

$$p(x) = \frac{\prod_{C\in\mathcal{C}} p_C(x_C)}{\prod_{S\in\mathcal{S}} p_S(x_S)}, \tag{2.1}$$

where p_C, the marginal of p over the clique C is Q_C and p_S is the marginal of any of the cliques in which the separator S is included.

This theorem allows to calculate the join probability distribution with a simple factorisation. of the distribution. It also implies that we don't have to use the conditional distribution, instead we can use the join probability distribution.

2.3 Conditional Gaussian networks

In this section we review the theory of Conditional Gaussian Network [Pérez2010, Bøttcher2004]. Conditional Gaussian Networks are mixed Bayesian networks, i.e., they contain both continuous and discrete variables.

Each continuous variable in the network follows a conditional Gaussian distribution given its parents. While each discrete variable follows a multinomial distribution, given their parents. Our main interest is using conditional Gaussian networks for classification tasks.

When performing classification with a conditional Gaussian network each of the continuous variables of the network, which are also called predictor variables, corresponds to each of the features of a given instance to be classified. On the other hand, there is only a discrete variable, which corresponds to the class of the instance, it follows a multinomial distribution and is a parent of every predictor variable.

This structure where all the predictor variables are sons of the class variable allows the factorization of the join probability. Suppose that each instance corresponds to a class C and has some features corresponding to

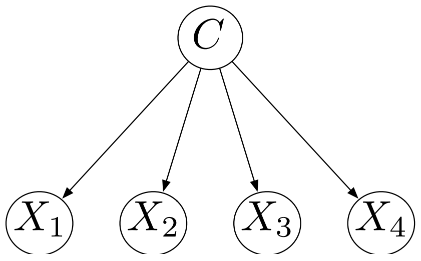

Figure 2.3: Naive Bayes structure, with four predictive variables.

predictor variables X, then the join probability can be factorized as

$$P(X, C) = P(X|C)P(C) = P(C)\prod_{i=1}^{n} P(X_i|Pa_i \cap X, C). \qquad (2.2)$$

As we are performing classification we are more interested in the probability of each class given some value for the predictor variables, this is $P(C|X)$, which can be calculated using the Bayesian rule.

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \propto P(X|C)P(C) \qquad (2.3)$$

Notice that $P(X)$ is exactly the same for all the possible classes, so it does not need to be calculated. The combination of equation 2.2 and 2.3 give us enough information to classify an instance.

Example 1. Naïve Bayes example is the most common and simple conditional Gaussian network. The assumption behind the model is that all the predictor variables are independent between them given the class, the bayesian network produced by this assumption can be seen in figure 1.

2.4 Gaussian Join tree classifiers

In the previous section we have reviewed conditional Gaussian networks introduced in [Bøttcher2004].We introduce now a family of models based on join trees structures and a Markov network which are equivalent to Conditional Gaussian Networks with an equivalent directed structure.

In Gaussian join tree classifiers the structure is represented as a join tree. We formally define join trees following [Cowell et al.1999].

Definition 14. A clique tree $\mathcal{T} = \{T_1, \ldots, T_n\}$ is a tree where each node $T_i \subseteq X$ is a set of variables. If T_i is a parent of T_j, the separator between

T_i and T_j is the set of variables $S_{ij} = T_i \cap T_j$. A join tree $\mathcal{T}$ is a clique tree such that for every pair of nodes T_i, T_j, $T_i \cap T_j$ is a subset of every separator on the unique path from T_i to T_j.

A Gaussian join tree classifier is a mixed network formed by a discrete variable corresponding to the class,which is the parent of all other variables. And a set of continuous variables that formed a Markov network with a structure of a join tree.

The selection of the join tree structure for the continuous variables allows us to use the decomposition explained in section 2.2.4. The factorization of the probability $P(C|X)$, where C is the class and X the predictive variables is easily done. Then the probability can be calculated as in equation 2.3, where the $P(X|C)$ is factorized as in 2.1.

Chapter 3

Learning Probabilistic Graphical Models

In this chapter we take a look to the problem of learning a model from data. The learning problem can be divided in two parts, structural learning and parameter learning. Structure learning consist in infering from data the conditional dependencies between the different variables of the model. Parameter learning consist in, once given a factorization of the model, learn the parameters of the different distributions of the factorization. From now on, we will focus on parameter learning.

We can divide the learning algorithms in two big families, frequentist and bayesian learning algorithms. We will consider two different algorithms: maximum likelihood, which is a frequentist approach and Bayesian model averaging, which is a Bayesian approach.

In this chapter we will start by reviewing general maximum likelihood in section 3.1. In section 3.2 we review Bayesian model averaging and shortly review the necessary theory to perform it over decomposable graphs.

3.1 Maximum likelihood

Maximum likelihood is one of the most used approaches to learn the parameters of a probability distribution. Even though it is one of the most used strategies it also presents some troubles like model over fitting.

If we consider a model with a set of parameters Λ and a dataset $\mathcal{D}$ of instances. Maximum likelihood approach estimates the value of the parameters by using those which maximize the likelihood.

$$P(\mathcal{D}|\Lambda)$$

The strategy of maximizing the likelihood, as we have said before can over fit the model, specially when the data is scarce. For example imagine that we are modelling the toss of a fair coin, and all our data consist in three consecutive flips in which we have obtained three heads. Then the maximum likelihood method infers that the probability of throwing the coin and obtain head is 1. Which is completely false and the worst possible approximation to the real model.

In order to obtain better model, and solve the problem of model over fitting we introduce in the next section the Bayesian model averaging method.

3.2 Bayesian model averaging

Bayesian model averaging tries to overcome the problem of model over fitting through considering a prior distribution over the space of parameters [Bishop2006]. This prior represents how probably we think a certain parameter is. The Bayes theorem gives us a way to learn a posterior distribution for the parameters from data.

Bayes theorem is one of the most important results of the probability theory. The main result is to calculate the probability of an event A given the event B by turning around the condition.

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}.$$

If we think in A as the parameters of the model and B the data we have the next result:

$$P(\Lambda|\mathcal{D}) \propto P(\mathcal{D}|\Lambda)P(\Lambda).$$

which means that the posterior distribution is proportional to the product of the likelihood of the data and the prior distribution.

When doing inference using a Bayesian model averaging method, we have to integrate over the whole space of parameters. This integration is not always possible and is required to use a sampling method such as Markov Chain Monte Carlo methods.

One usual problem with this approach is that most of the time the most important factor when choosing a prior distribution is their mathematical convenience rather than the information they take.

3.2.1 Hyper Markov distributions over decomposable graphs

We have seen in Section 2.2.4 that a probability distribution can be easily factorized in term of the cliques and separators if the associated graph is decomposable. In order to perform Bayesian model averaging over this models we need to introduce hyper Markov distributions.

In this section we sufficiently review the fundamental results in [Dawid and Lauritzen1993]. The interested reader can find some of the missing definitions and additional details in [Dawid and Lauritzen1993].

Let θ be a quantity parametrising a graphical model $M(\mathcal{G})$. A hyper Markov law is then defined by a property which mimics the global Markov property, at the parameter level

Definition 15. A law on $M(\mathcal{G})$ is called (weak) hyper Markov over $\mathcal{G}$ if for any decomposition (A, B) of $\mathcal{G}$

$$\theta_A \perp\!\!\!\perp \theta_B | \theta_{A \cap B}$$

Definition 16. A law on $M(\mathcal{G})$ is called strong hyper Markov over $\mathcal{G}$ if for any decomposition (A, B) of $\mathcal{G}$

$$\theta_{B|A} \perp\!\!\!\perp \theta_A$$

Strong hyper Markov laws produce an especially simple decomposition of the Bayesian analysis into a collection of sub-analyses for smaller problems. Thus, the posterior after observing some data can be assessed locally.

The following two propositions from [Dawid and Lauritzen1993] will be needed later. They give sufficient conditions to prove that a given probability law is strong hyper Markov.

Proposition 3. *Given a set of hyperconsistent laws $\{\mathcal{M}_C\}$ over clique marginals, there is a unique hyper Markov law over $\mathcal{G}$ satisfying those marginals, which is called the hyper Markov combination of $\{\mathcal{M}_C\}$.*

Proposition 4. *Let $\mathcal{P} \subseteq M(\mathcal{G})$ be a subfamily of the Markov models over $\mathcal{G}$. Assume that $\mathcal{P}$ is weak meta Markov, and for any complete set S in $\mathcal{G}$ the model $\mathcal{P}_S$ form a full exponential family. Let $\mathcal{L}$ be a hyper Markov law such that, for any clique C, the law of θ_C is a conjugate prior distribution for the model $\mathcal{P}_C$. Then $\mathcal{L}$ is strong hyper Markov.*

Chapter 4

Learning CGN by ML

We have already talk about learning in Chapter 3, here we focus on learning conditional Gaussian networks using maximum likelihood. We start by reviewing multinomial and Gaussian distribution, how to learn their parameters and how to calculate their conditional distribution. Finally we put everything together and get into how to learn conditional Gaussian Networks

4.1 The multinomial distribution

The multinomial distribution is a probability distribution over discrete probabilities that can take two or more different values. It only has an array of parameters where each parameter is the probability that the variable take. The array of parameters $c = (c_1, \ldots, c_K)$ full fill the following properties.

$$c_i > 0$$
$$p(x = k|c = c_k = 1$$

4.1.1 Parameter learning

Parameter of the multinomial distribution are easily learned by maximum likelihood. The value of each parameter c_i is just proportion of times that the variable takes the i-th value in the data. In other words if m_i is the number of times that the variable takes the value i in the data, and the data has n instances, then

$$c_i = \frac{m_i}{n}.$$

4.2 The normal distribution

The normal distribution, also known as Gaussian distribution, is defined on R^k. It has two parameters: the mean μ which is a real-valued vector, and the covariance matrix Σ wich is a positive definite $k \times k$ matrix. In the literature the covariance matrix can be substituted by its inverse, the precission matrix. The probability density function is

$$\mathcal{N}(X|\mu, \Sigma) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} \exp\left(\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$

4.2.1 Parameter learning

Paremeters of normal distribution can be learned from data by different approaches. The most common one is to use maximum likelihood estimations. Given a dataset $D\,\{x_i\}_{i=1}^n$ with $x_i \in R^k$ the maximum likelihood estimator of the mean vector is

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{4.1}$$

The estimation of S can be calculated as

$$S = \left[\frac{1}{n}\sum_{l=0}^{n}(x_l^i - \bar{x}^i)(x_l^j - \bar{x}^j)\right]_{0<i,j\leq k}, \tag{4.2}$$

where x^i is the ith entry of the vector x.

Instead of using maximum likelihood strategies for learning parameters we could use maximum a posteriori learning methods. For this we have to define a prior distribution over the space of parameters of the distribution. In next section we introduce Normal Inverse Wishart, as a prior for the normal distribution.

4.2.2 Condtional Gaussian distribution

Given a multidimensional variable $z = (z_1, \ldots, z_k)$ which follows a normal distribution with parameters μ and Σ we can calculate the conditional distribution of a subset of this,say x variables given the other, y. Without loss of generality we can think that x are the first l variables, with l smaller than k. We have the following decomposition of the parameters:

$$\mu = (\mu_x, \mu_y)$$
$$\Sigma = \begin{pmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_y \end{pmatrix}$$

Then we can calculate the conditional distribution of $P(z_x|z_y)$ which is also a Gaussian distribution with parameters

$$\begin{aligned}\mu_{x|y} &= \mu_x + \Sigma_{x,y}\Sigma_y^{-1}(z_y - \mu_y),\\ \Sigma_{x|y} &= \Sigma_x - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}.\end{aligned}$$

4.3 Learning conditional Gaussian Networks

Now we have seen how can we learn the different distributions separately, now we focus on how to learn all the parameters of a conditional Gaussian Network.

Algorithm 1 explains how to learn the parameters of the network from a dataset D. Basically it can be split in two parts, the first one is to determine the probability of each class, the second to learn, for each class a different distribution for each variable.

Algorithm 1 Maximum Likelihood learning of CGN parameters

```
function LearnCGNParameters($\mathcal{X}, \mathcal{D}$)
    for each class $c$ do
        Set $P(C = c)$ as the proportion of instances of class $c$ in $\mathcal{D}$
        for each node $X_i \in \mathcal{X}$ do
            Calculate the mean and the covariance probability of the join distribution of $X_i$ and their predictor parents, according to equation 4.1 and 4.2.
            Calculate the conditional Gaussian distribution of $X$ given their parents as explained in Section 4.2.2.
        end for
    end for
end function
```

Other possible strategies to learn parameters are possible. We will focus in the comparison between Bayesian Model Averaging and Maximum Likelihood. In the following chapters we will give the sufficient tools to use Bayesian Model Averaging over decomposable graphs.

Chapter 5

Learning Gaussian Join Tree classifiers by BMA

In the previous chapter we have focus on learning a conditional Gaussian network using a Maximum likelihood. As we have commented before, this model can lead to model over fitting, specially when data is scarce. This problem can be solved by Bayesian learning techniques. In this chapter we focus on the use of Bayesian Model Averaging to learn the parameters of a Gaussian Join Tree distribution. We start by reviewing the Dirichlet prior for the multinomial and introducing the Hyper Inverse Wishart Distribution. Finally we give the algorithms to learn the parameters of a Gaussian Join Tree classifier.

5.1 Dirichlet distribution

Dirichlet distribution is a conjugate prior for the multinomial distribution [Bishop2006].The Dirichlet distribution is defined as follows.

$$P(\mathbf{c}|\alpha) = \frac{\Gamma({}_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}} \prod_{k=1}^{K} c_k^{\alpha_k - 1}$$

Where k is the dimension of the distribution, $c_k > 0$, $\sum_{k=1}^{K} c_k = 1$, and $\Gamma(x)$ is the Gamma function. The parameter α_i must be an integer, and it represents how many times the class i has been saw in the prior. So, if according to our belief one class is more probable it should translate in a larger value for its corresponding α parameters in the prior.

5.1.1 Conjugancy

Since the Dirichlet distribution is conjugate to the multinomial distribution we can learn the posterior distribution by updating the parameters of the prior. Given a dataset $\mathcal{D}$, the parameters α' of the posterior distribution are updated by adding the counts of each class, i.e., let n_i be the number of instances in $\mathcal{D}$ corresponding to the class i then

$$\alpha'_i = \alpha_i + n_i.$$

5.1.2 Predictive posterior distribution

Inference when using the Dirichlet distribution as a prior for the multinomial distribution can be calculated exactly. The posterior predictive distribution is obtained by integrating out the parameter c_i.

$$P(X|\alpha') \int_{\mathbf{c}} \mathcal{MN}(X|c)\mathcal{D}(c|\alpha') \propto \alpha'$$

Then, the probability of a given class is the proportion of the class in the data, and in the prior. In other words, the predictive posterior distribution is another multinomial distribution with parameters $\frac{\alpha'}{\sum_{i=1}^{K} \alpha'_i}$.

5.2 Hyperinverse Wishart distribution

Now we return to Hypermarkorv theory, and introduce the hyperinverse Wishart distribution in order to introduce later the Hyper normal inverse Wishart distribution, which will be used as a prior for the Gaussian distribution over a decomposable graph.

Let $\mathcal{G}$ be a decomposable graph over a set of continuous variables. We are interested in the subfamily of models which are in $M(\mathcal{G})$ and which are assumed to be jointly multivariate normal with mean equal to zero and unknown positive definite covariance matrix Σ, that is $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \Sigma)$. For this particular family, it is possible to define a strong hyper Markov distribution. It was introduced in [Dawid and Lauritzen1993], under the name of hyper inverse Wishart distribution

Definition 17. Let $\Phi^{\mathcal{C}} = \{\Phi^C, C \in \mathcal{C}\}$ be a collection of positive definite dispersion matrices, where Φ^C is the dispersion matrix over the variables in clique C. Assume that the matrices are consistent, that is, for each $B \subseteq$

$C_1 \cap C_2$, the sub-matrices[1] $\Phi^{C_1}[B, B]$ and $\Phi^{C_2}[B, B]$ are identical. Let δ be a positive real number. We can define a collection of hyper consistent Markov laws by defining over each matrix Σ^C the following law:

$$\mathcal{L}(\Sigma^C) = \mathcal{IW}(\delta; \Phi^C).$$

The law that results from the hyper Markov combination of this collection of laws is called hyper inverse Wishart and noted $\mathcal{HIW}(\delta, \Phi^{\mathcal{C}})$.

The $\mathcal{HIW}$ is proved strong hyper Markov in [Dawid and Lauritzen1993].

5.3 Hyper normal inverse Wishart distribution

$\mathcal{HIW}$ laws can only be used when the multivariate mean is known to be $\mathbf{0}$. We are interested in the more general subfamily of models in $M(\mathcal{G})$ with any possible mean, that is $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$. We introduce the hyper normal inverse Wishart distribution and prove that it is strong hyper Markov.

Definition 18. Let $\Phi^{\mathcal{C}} = \{\Phi^C, C \in \mathcal{C}\}$ be a collection of matrices as in definition 17. Let $\boldsymbol{\mu}^{\mathcal{C}} = \{\boldsymbol{\mu}^C, C \in \mathcal{C}\}$ be a collection of vectors such that for each $B \subseteq C_1 \cap C_2$, $\boldsymbol{\mu}^{C_1}[B] = \boldsymbol{\mu}^{C_2}[B]$. Let κ and δ be non-negative real numbers. Since the marginal of a normal inverse Wishart depends only on the corresponding sub-vector and sub-matrix (see section A.2.3), the collection of laws

$$\mathcal{L}(\Sigma^C) = \mathcal{NIW}(\boldsymbol{\mu}^C, \kappa, \delta, \Phi^C)$$

is hyper consistent. The hyper Markov combination of its elements is called hyper normal inverse Wishart and noted $\mathcal{HNIW}(\boldsymbol{\mu}^{\mathcal{C}}, \kappa, \delta, \Phi^{\mathcal{C}})$. By proposition 3 the $\mathcal{HNIW}$ law is hyper Markov.

Theorem 2. *The $\mathcal{HNIW}$ distribution is strong hyper Markov.*

Proof. Is a direct application of proposition 4. First, note that in our case, the set of models is the set of multidimensional Gaussian models that factorize over $\mathcal{G}$. Thus, it is weak meta Markov and over a complete set, it forms a full exponential family. Since the $\mathcal{HNIW}$ law is hyper Markov, and for any clique C, it is a conjugate distribution for the multidimensional Gaussian over C, by proposition 4 it is strong hyper Markov. $\square$

[1]Given a matrix M and sets of indexes I,J, we note $M[I, J]$ the sub-matrix that keeps the rows with indexes in I and the columns with indexes in J. Equivalent notation is used for vectors.

5.3.1 Conjugacy

Since the $\mathcal{HNIW}$ is strong hyper Markov, we can update each of the marginal distributions locally. Furthermore, since the local laws are $\mathcal{NIW}$ the update can be done using the result in appendix A.2.1. Hence, in order to learn from data we only have to update our hyperparameters using the following equations.

$$\begin{aligned}
\boldsymbol{\eta}' &= \frac{\kappa\boldsymbol{\eta} + n\overline{\mathbf{x}}}{\kappa + n}, && (5.1)\\
\kappa' &= \kappa + n, && (5.2)\\
\delta' &= \delta + n, && (5.3)\\
\Psi' &= \Psi + (n-1)S + \frac{\kappa n}{\kappa + n}(\overline{\mathbf{x}} - \boldsymbol{\eta})(\overline{\mathbf{x}} - \boldsymbol{\eta})^T. && (5.4)
\end{aligned}$$

5.3.2 Predictive distribution

The predictive distributions for each clique are multivariate t distributions, as given by the equation:

$$\begin{aligned}
p(\mathbf{x}|\boldsymbol{\eta}, \kappa, \delta, \Psi) &= \\
&= \int_{\boldsymbol{\mu},\Sigma} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \Sigma)\mathcal{NIW}(\boldsymbol{\mu}, \Sigma|\boldsymbol{\eta}, \kappa, \delta, \Psi) = \\
&= t_\delta(\boldsymbol{\eta}, \frac{\kappa+1}{\kappa\delta}\Psi). \quad (5.5)
\end{aligned}$$

To get the distributions over a separator, we marginalize the distribution of one of the cliques that it separates using the result provided in section A.1.1 about the marginal of a multivariate t.

5.4 Gaussian Join Tree Classifiers

We have already introduced Gaussian Join Tree Classifiers in section 2.4. In this section we introduce how to learn the parameters of a Gaussian Join tree classifier using Bayesian model averaging. The parameters of a GJT classifier are determined combining the results described in section 5.3.1 and 5.3.2. For each of the classes, each of the cliques and separators will be assigned a multivariate t distribution. The algorithmic details are provided in Algorithmd 3, 4 and 5.

In Algorithm 3 we start setting the prior distributions by specifying their parameters, in line 1 to learn the parameters of the multinomial distribution

Algorithm 2 General GJT classifier

```
function GJTLearner($\mathcal{D}$)
    $\mathcal{T} := DetermineCliqueTree(\mathcal{D})$
    $\Theta := LearnGJTParameters(\mathcal{T}, \mathcal{D})$
    return $\langle \mathcal{T}, \Theta \rangle$
end function
```

Algorithm 3 Bayesian learning of GJT parameters

```
1: function LearnGJTParameters($\mathcal{T}, \mathcal{D}$)
2:     $LearnClassesByBMA(\mathcal{D})$
3:     $\mathcal{C} := Cliques(\mathcal{T})$
4:     for each class $c$ do
5:         $LearnCliquesByBMA(c, \mathcal{C}, \mathcal{T})$
6:
```

for the classes of the model as explained in algorithm 4. Then, from line 3 to 5, we learn the parameters of the cliques for each different class according to algorithm 5.

Algorithm 4 Bayesian learning of class parameters

```
1: function LearnClassesByBMA($\mathcal{D}$)
2:     Set the parameters of prior distribution $\alpha$.
3:     Asses the number of data of each class $n_i$ from data $\mathcal{D}$.
4:     $\alpha' := \alpha + \mathbf{n}$
5:     Normalize the vector $\alpha'$.
6:     for each class $c$ do
7:         Set the probability of $c$ equal to it corresponding entry of $\alpha'$.
8:     end for
```

In algorithm 4 we learn the probabilities of the different classes. In line 1 we set the parameters of the Dirichlet prior. In lines 2-3 we update the parameters and in the last line we set the posterior predictive distributions of the classes.

Algorithm 5 explains how to learn the parameters of gaussian distribution over a set of cliques, it basically iterates over the set of cliques and calculate all the distribution and then marginalizes the distribution over the separators. In line 3 we asses the prior distribution by assessing its parameters, next, we calculate the empirical mean and the covaraince matrix. From line 6 to line 9 we update the parameters of the prior distribution to calculate the posterior. Finally we define a t distribution over the clique and marginalize it over the separators formed by the intersection between the clique and their sons in the clique tree structure.

Algorithm 5 Bayesian learning of cliques parameters

```
1: function LearnCliquesByBMA($c, \mathcal{C}, \mathcal{T}$)
2:     for $C \in \mathcal{C}$ do
3:         Set the parameters of the prior distribution $\kappa$, $\delta$,$\boldsymbol{\eta}$, and $\Psi$.
4:         With the data from class $c$ assess the number of data points $n(c)$, the sample mean $\overline{\mathbf{y}}(c)$
5:         and the sample covariance matrix $\Sigma(c)$.
6:         $\kappa'(c) := \kappa(c) + n(c)$
7:         $\delta'(c) := \delta(c) + n(c)$
8:         $\boldsymbol{\eta}' := \frac{\kappa(c)\boldsymbol{\eta}[C]+n(c)\overline{\mathbf{x}}(c)[C]}{\kappa(c)+n(c)}$
9:         $\Psi' := \Psi[C,C] + (n(c) - 1)\Sigma(c)[C,C] + \frac{\kappa n(c)}{\kappa+n(c)}(\overline{\mathbf{x}}(c)[C] - \boldsymbol{\eta}[C])(\overline{\mathbf{x}}(c)[C] - \boldsymbol{\eta}[C])^T$.
10:        $d(C,c) := t_{\delta'}(\boldsymbol{\eta}', \frac{\kappa'+1}{\kappa'\delta'}\Psi')$
11:        for each clique $C'$ child of $C$ in $\mathcal{T}$ do
12:            $d(C' \cap C, c) = t_{\delta'}(\boldsymbol{\eta}'[C' \cap C], \frac{\kappa'+1}{\kappa'\delta'}\Psi'[C' \cap C, C' \cap C]])$
13:        end for
14:    end for
15:
```

5.5 Predicting

Given a new unclassified data point, Bayes formula is used to determine the posterior probability for each class. We use Laplace formula to assess the prior probability for class c (which is equivalent to assuming a Dirichlet prior) and get the posterior by multipling it by joint probability obtained using equation (2.1).

Chapter 6

Empirical Comparison

In this section we compare GJT classifiers with classifiers making similar independency assumptions, but whose parameters are adjusted by maximum likelihood, namely CGN classifiers, as proposed in [Pérez2010]. We use two different datasets from the bioinformatics domain, one for ovarian cancer and one for pancreatic cancer. Both datasets have been obtained from the NIH and contain high resolution spectrograms coming from surface-enhanced laser desorption/ionization time of flight mass spectrometry (SELDI-TOF MS). In both datasets the objective is to distinguish spectrograms coming from cancer patients from those coming from control individuals.

6.1 Spectogram datasets

6.1.1 Ovarian cancer dataset

The ovarian cancer dataset contains a total of 216 spectrograms, 121 from cancer patients and 95 controls. The m/z values do not coincide along the different spectrograms. Thus, to create the variables, the m/z axis data has been discretized into different bins, creating a variable for each bin, for a total of 11300 variables. Thus, the number of variables largely exceeds the number of data points. For each spectrogram, the average of the values of each bin has been assigned to that bin's variable.

6.1.2 Pancreatic cancer dataset

The pancreatic cancer dataset contains a total of 181 spectrograms, 101 from cancer patients and 80 controls. Each spectrogram is defined by 6771 variables which in this case are aligned, so no discretization is needed.

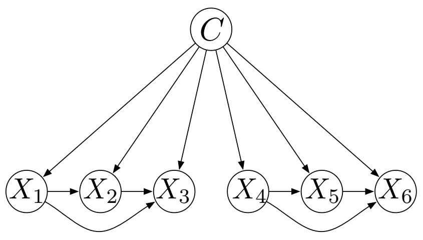

Figure 6.1: Structure of the graph for k-BOX model

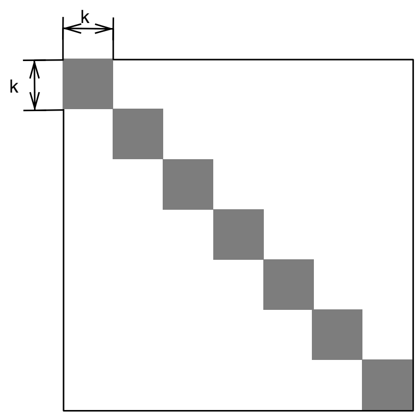

Figure 6.2: Structure of the covariance matrix for k-BOX model

6.2 Dependency structures

We have used two different families of structures. Both are based on the hypothesis that those variables that represent close m/z relations are more likely to have large correlations than those whose m/z values are further away.

6.3 k-BOX structure

A k-BOX structure can be defined over an ordered set of variables $V = \langle X_1, \ldots, X_n \rangle$. It is a restriction of *semi Naïve Bayes* [Larrañaga et al.2006], also known as *JAN* [Pérez2010]. In a JAN structure the variables are joined in groups to form multivariate distributions. In the k-BOX structure we divide the variables in disjoints sets of k contiguous variables. The network structure can be seen in Figure 6.3 and the corresponding covariance matrix in Figure 6.3. In our case the ordering is provided by the m/z value.

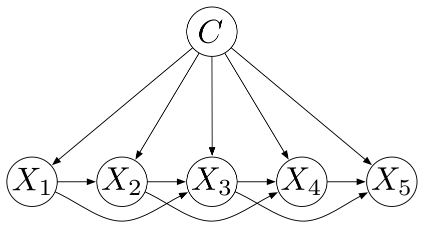

Figure 6.3: Structure of the graph for k-BAND model

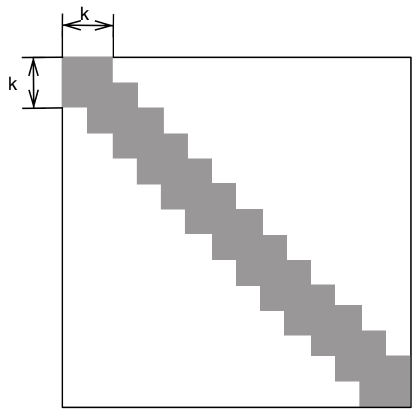

Figure 6.4: Structure of the covariance matrix for k-BAND model

6.3.1 k-BAND structure

The second proposed structure is the k-BAND structure. In k-BAND, we assume that each variable is independent of all the remaining variables given the $k-1$ variables that precede it and the class variable. Therefore, the k-BAND structure is formed by cliques of size k with separators of $k-1$ variables.

The covariance matrix for a k-BAND structure is a band of size k around the diagonal, as is shown in Figure 6.3.1. An example of the structure is shown in Figure 6.3.1.

6.4 The prior

As we are working with Bayesian model average we need to define a prior distribution for both, the multinomial and the Gaussian distributions. In the case of multinomial distribution we have set as prior a Dirichlet distribution with $\alpha_i = 1$ for $i = 1, 2$. This prior suppose that we know that exist instances of both classes, but we don't risk to much to say in what proportion. In the

		Learning Method	
		Maximum Likelihood	Maximum a Posteriori
Structure	K-BOX	CGN-K-BOX	GJT-K-BOX
Model	K-BAND	CGN-K-BAND	GJT-K-BOX

Table 6.1: Summary of combination of methods and structures.

case of the Gaussian we choose a Normal Inverse Wishart prior distribution which uses the following parameters. $\eta = 0$, $\delta = \kappa = 1$ and Ψ equal to the identity matrix, what includes a belief on the independence of the variables.

6.5 The experiment

We have performed different comparisons for each dataset. We have used 4 different classifier methods. Each classifier method is the combination of a structure and a learning principle. The different combinations are shown in table 6.5.

For cancer datasets the methods have been run with different value for K. For ovarian cancer dataset K takes values from 1 to 50, and for pancreatic cancer dataset the methods have been run with values of K from 1 to 30.

A 10-cross fold validation has been performed for each method. In other words, the dataset have been divided in 10 different subsets. We have classified each of this subsets while the model has been learn with the other 9 subsets.

6.5.1 Measures

Two measures have been used to compare the performance of the classifiers. The accuracy is the proportion of corrected classified patients. We can understand accuracy as a measure of how well we classify.

$$\text{Accuracy} = \frac{\text{corrected classified instances}}{\text{instances}}$$

We also use conditional log-likelihood (CLL), which is the log-probability assigned to the correct class of the instances.CLL measures how accurately the probabilities for each class are estimated, which is very relevant for adequate decision making. If c_i is the class associated to an instances x_i, the cll is defined as

$$CLL = \sum_{i=0}^{m} \log(P(C = c_i|x_i))$$

6.6 Results

We have run a sequence of experiments on each dataset to compare the different structures (k-BOX and k-BAND) and parameter learning methods (CGN and GJT) varying the k parameter.

The source code used to perform the experiments is available at `http://www.iiia.csic.es/~cerquide/pypermarkov`

6.6.1 Ovarian data

The ovarian data has been modelled using k-BOX and k-BAND structures with k ranging from 1 to 50. In Figure 6.5 we show the mean accuracy versus the number of parameters in the model and in Figure 6.6 we show the mean CLL versus the number of parameters in each structure. We see that in that dataset, k-BAND models are more accurate that k-BOX models. For low values of k, CGN performs better than GJT, but GJT has a largest accuracy and a highest CLL at its peak, and shows a more graceful decay as the number of parameters grows beyond that peak.

6.6.2 Pancreatic data

The pancreatic cancer data has been classified using k-BOX and k-BAND structures with k ranging from 1 to 30. Figure 6.7 and Figure 6.8 show respectively the mean accuracy and mean CLL against the number of parameters. Note that the accuracy for this dataset is much lower and close to the frequency of the largest class, that is 55.8% in this datasets. Previous studies have shown that the accuracy results for this dataset are much lower than for the previous one. The k-BOX model using GJT appears to reach the highest accuracy and the k -BAND model reaches the highest CLL also for pancreatic cancer.

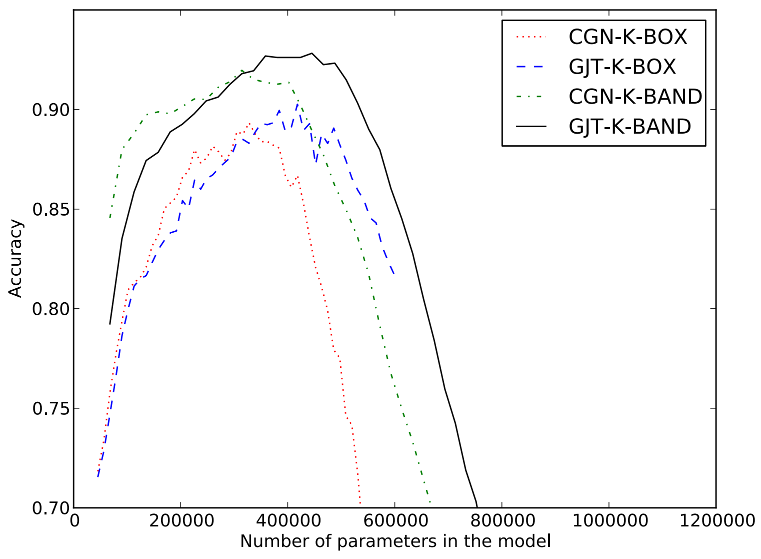

Figure 6.5: Prediction of ovarian cancer. Accuracy versus number of parameters in model.

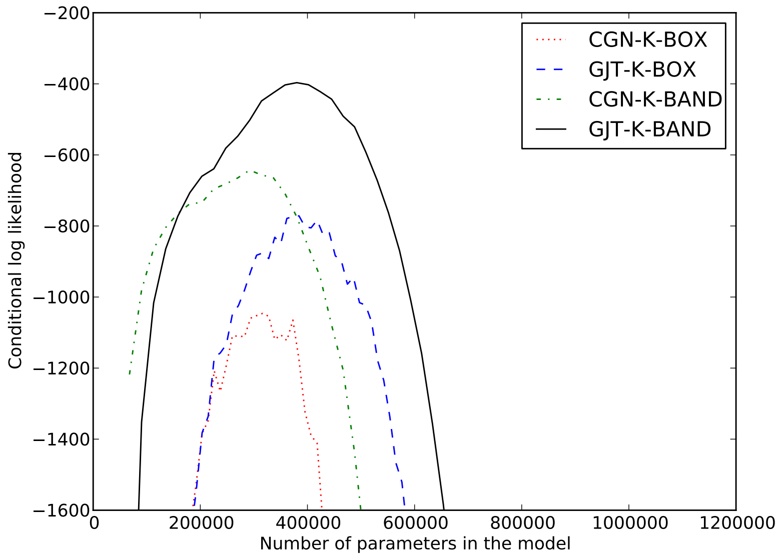

Figure 6.6: Prediction of ovarian cancer. CLL versus number of parameters in model.

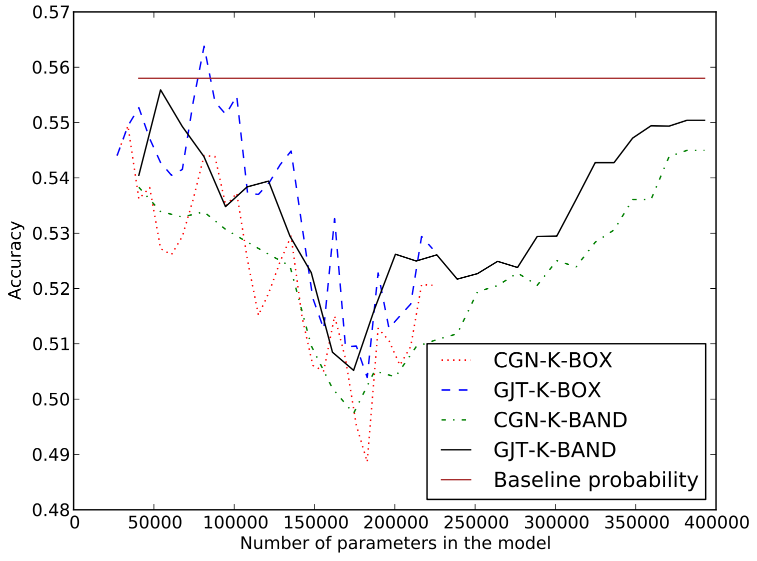

Figure 6.7: Prediction of pancreatic cancer. Accuracy versus number of parameters in model.

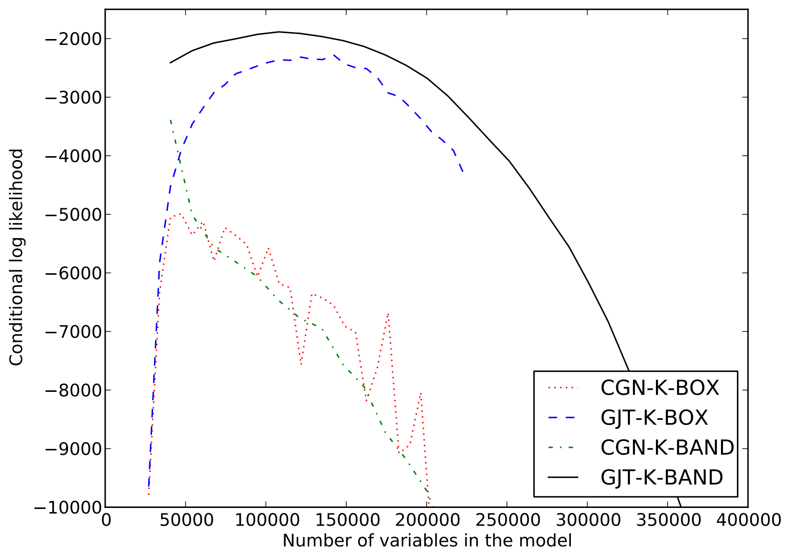

Figure 6.8: Prediction of pancreatic cancer. CLL versus number of parameters in model.

Chapter 7

Conclusions and Future Work

We have introduced a new family of classifiers for continuous domains, namely Gaussian join Tree classifiers that perform exact Bayesian averaging over the parameters by virtue of the hyper normal inverse Wishart law, that we have introduced and proved to be strong hyper Markov. To assess the benefits we have compared GJT classifiers with GCN classifiers which assuming the same set of independences adjust parameters using maximum likelihood. We performed our comparison with two high resolution mass spectrometry datasets for ovarian and pancreatic cancer prediction. We have seen that, for two simple dependency structures, our classifiers reach a better peak accuracy and a consistently better conditional log likelihood.

We have introduced k-BOX and k-BAND structure, which are part of the same family of join trees structures. For example different sizes of the separators define different models in the family. The study of this family remains also as future work.

The GJT parameter averaging can be performed over any set of dependencies that can be encoded into a decomposable graph. We can adapt any algorithm for learning the structure of a CGN classifier so that it outputs a join tree by moralizing and triangulating the DAG that encodes the structure of the CGN, and then running maximum cardinality search. A future line of work is comparing along the datasets in [Pérez2010] using the same structure learning algorithms that they suggest, thus testing if the benefits extend to domains with a smaller number of attributes.

Finding a structure learning method for large Bayesian networks remains as future work. More informed dependencies structures should imply better results in mass spectra classification.

Bibliography

[Bishop2006] C.M. Bishop. 2006. *Pattern recognition and machine learning.* Springer New York.

[Bøttcher2004] Susanne Gammelgaard Bøttcher. 2004. *Learning Bayesian Networks with Mixed Variables.* Ph.D. thesis, Aalborg University.

[Buntine1996] Wray Buntine. 1996. A Guide to the Literature on Learning Probabilistic Networks from Data. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 8(2):195–209.

[Cowell et al.1999] Robert G. Cowell, A. Philip Dawid, Steffen L. Lauritzen, and David J. Spiegelhalter. 1999. *Probabilistic Networks and Expert Systems.* Springer-Verlag.

[Dawid and Lauritzen1993] A. P. Dawid and S. L. Lauritzen. 1993. Hyper Markov Laws in the Statistical Analysis of Decomposable. *The Annals of Statistics*, 21(3):1272–1317.

[Elo and Schwikowski2011] Laura L. Elo and Benno Schwikowski. 2011. Mining proteomic data for biomedical research. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(February):n/a–n/a, August.

[Geiger and Heckerman1994] Dan Geiger and David Heckerman. 1994. Learning gaussian networks. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-94)*, pages 235–243.

[Gelman et al.2004] Andrew Gelman, JB Carlin, HS Stern, and Rubin. 2004. *Bayesian data analysis.*

[III et al.2002] Emanuel F Petricoin III, Ali M Ardekani, Ben A Hitt, Peter J Levine, Vincent A Fusaro, Seth M Steinberg, Gordon B Mills, Charles Simone, David A Fishman, Elise C Kohn, and Lance A Liotta. 2002. Use of proteomic patterns in serum to identify ovarian cancer. *The Lancet*, 359(9306):572 – 577.

[Koller and Friedman2009] D. Koller and N. Friedman. 2009. *Probabilistic graphical models: principles and techniques.* The MIT Press.

[Kotz and Nadarajah2004] Samuel Kotz and Saralees Nadarajah. 2004. *Multivariate T-Distributions and Their Applications.* Cambridge University Press, Cambridge.

[Larrañaga et al.2006] Pedro Larrañaga, P, Aritz Pérez, Iñaki Inza, and Pedro Larra. 2006. Supervised classification with conditional Gaussian networks : Increasing the structure complexity from naive Bayes. *International Journal of Approximate Reasoning*, 43(January):1–25.

[Lauritzen1996] Steffen L. Lauritzen. 1996. *Graphical models.* Oxford University Press.

[Pérez2010] Aritz Pérez. 2010. *Supervised classification in continuous domains with Bayesian networks.* Ph.D. thesis, Universidad del Pais Vasco.

Appendix A

Distributions

A.1 Multivariate t-distributions

A p-dimensional random vector $\mathbf{x}$ is said to have the p-variate t distribution with degrees of freedom ν, mean vector μ , and correlation matrix R (that is $x \sim t_\nu(\mu, R)$) if its joint pdf is given by

$$p(\mathbf{x}|\nu,\mu,R) = \frac{\Gamma((\nu+p)/2)}{(\pi\nu)^{p/2}\Gamma(\nu/2)|R|^{1/2}} \cdot \\ \cdot [1 + \frac{1}{\nu}(x-\mu)R^{-1}(x-\mu)]^{-(\nu+p)/2} \quad \text{(A.1)}$$

A.1.1 Marginals

Let $\mathbf{x}$ be a p-dimensional random vector $\mathbf{x} \sim t_\nu(\boldsymbol{\mu}, R)$. Furthermore let $\mathbf{x} = (\mathbf{x_1}, \mathbf{x_2})$ where $\mathbf{x_1}$ is p_1 dimensional, $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2)$ and $R = \begin{pmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{pmatrix}$. We have that

$$\mathbf{x_1} \sim t_\nu(\boldsymbol{\mu}_1, R_{11}) \quad \text{(A.2)}$$

For more information regarding multivariate t-distributions see [Kotz and Nadarajah2004].

A.2 Normal Inverse Wishart distributions

The inverse Wishart distribution is defined on real-valued positive-definite $p \times p$ matrices. It has two parameters: a real number $\delta > 0$ and a $p \times p$

positive-definite matrix Ψ. The probability density function is

$$\mathcal{IW}(X|\delta,\Psi) = \frac{|\Psi|^{\frac{\delta+p-1}{2}}|X|^{-\frac{\delta+2p}{2}}}{2^{\frac{(\delta+p-1)p}{2}}\Gamma_p(\frac{\delta+p-1}{2})}e^{-\frac{1}{2}tr(\Psi X^{-1})}$$

The normal inverse Wishart distribution is defined on pairs composed of (i) vectors of dimension p and (ii) real-valued positive-definite $p\times p$ matrices. It has four parameters: a p dimensional vector $\boldsymbol{\eta}$ that encodes the location, a positive real number κ that acts as scaling factor, and δ and Ψ as in the inverse Wishart. The probability density function is

$$\mathcal{NIW}(\boldsymbol{\mu},\Sigma|\boldsymbol{\eta},\kappa,\delta,\Psi) = \mathcal{N}(\boldsymbol{\mu}|\boldsymbol{\eta},\frac{1}{\kappa}\Sigma)\cdot\mathcal{IW}(\Sigma|\delta,\Psi)$$

A.2.1 Conjugacy

The normal inverse Wishart distribution is conjugate to the multivariate normal [Gelman et al.2004]. Thus, if we assume as prior a $\mathcal{NIW}(\boldsymbol{\mu},\Sigma|\eta,\kappa,\delta,\Psi)$ and we are given a sample X from a multivariate normal, the posterior will be a $\mathcal{NIW}(\boldsymbol{\mu},\Sigma|\eta',\kappa',\delta',\Psi')$ where

$$\begin{aligned}\boldsymbol{\eta}' &= \frac{\kappa\boldsymbol{\eta}+n\overline{\mathbf{x}}}{\kappa+n}, &\text{(A.3)}\\ \kappa' &= \kappa+n, &\text{(A.4)}\\ \delta' &= \delta+n, &\text{(A.5)}\\ \Psi' &= \Psi+(n-1)S+\frac{\kappa n}{\kappa+n}(\overline{\mathbf{x}}-\boldsymbol{\eta})(\overline{\mathbf{x}}-\boldsymbol{\eta})^T. &\text{(A.6)}\end{aligned}$$

and S is the sample covariance.

A.2.2 Predictive distribution

The predictive distribution is the result of integrate over the space of parameter the product of the normal distribution and the Normal Inverse Wishart. This operation, give as result a multivariate t distribution depending on the hyperparameters of the model.

$$\begin{aligned}&p(\mathbf{x}|\boldsymbol{\eta},\kappa,\delta,\Psi) =\\ &\quad= \int_{\boldsymbol{\mu},\Sigma}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\Sigma)\mathcal{NIW}(\boldsymbol{\mu},\Sigma|\boldsymbol{\eta},\kappa,\delta,\Psi) =\\ &\qquad= t_\delta(\boldsymbol{\eta},\frac{\kappa+1}{\kappa\delta}\Psi). \quad \text{(A.7)}\end{aligned}$$

A.2.3 Marginals

The next proposition give us the marginals of the Normal Inverse Wishart. This result will be necessary later, when we talk about hyper Markov distributions.

Proposition 5. *Let* $\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}$, $\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$, $\boldsymbol{\eta} = \begin{pmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \end{pmatrix}$ *and* $\Psi = \begin{pmatrix} \Psi_{11} & \Psi_{12} \\ \Psi_{21} & \Psi_{22} \end{pmatrix}$.

If $(\boldsymbol{\mu}, \Sigma) \sim \mathcal{NIW}(\boldsymbol{\eta}, \kappa, \nu, \Psi)$, *then* $(\boldsymbol{\mu}_1, \Sigma_{11}) \sim \mathcal{NIW}(\boldsymbol{\eta}_1, \kappa, \nu, \Psi_{11})$

Proof. The marginal can be assessed as follows

$$\begin{aligned}
&P(\boldsymbol{\mu}_1, \Sigma_{11} | \boldsymbol{\eta}, \kappa, \delta, \Psi) = \\
&= \int_{\boldsymbol{\mu}_2, \Sigma_{12}, \Sigma_{22}} \mathcal{NIW}(\boldsymbol{\mu}, \Sigma | \boldsymbol{\eta}, \kappa, \delta, \Psi) = \\
&= \int_{\boldsymbol{\mu}_2, \Sigma_{12}, \Sigma_{22}} \mathcal{N}(\boldsymbol{\mu} | \boldsymbol{\eta}, \frac{1}{\kappa}\Sigma) \cdot \mathcal{IW}(\Sigma | \delta, \Psi) = \\
&= \int_{\boldsymbol{\mu}_2, \Sigma_{12}, \Sigma_{22}} \mathcal{N}(\boldsymbol{\mu}_1 | \boldsymbol{\eta}_1, \frac{1}{\kappa}\Sigma_{11}) P(\boldsymbol{\mu}_2 | \boldsymbol{\mu}_1, \boldsymbol{\eta}, \kappa, \delta, \Psi) \cdot \\
&\quad \cdot \mathcal{IW}(\Sigma_{11} | \delta, \Psi_{11}) P(\Sigma_{22}, \Sigma_{21} | \Sigma_{11}, \delta, \Psi) = \\
&= \mathcal{N}(\boldsymbol{\mu}_1 | \boldsymbol{\eta}_1, \frac{1}{\kappa}\Sigma_{11}) \mathcal{IW}(\Sigma_{11} | \delta, \Psi_{11}) \cdot \\
&\cdot \int_{\boldsymbol{\mu}_2, \Sigma_{12}, \Sigma_{22}} P(\boldsymbol{\mu}_2 | \boldsymbol{\mu}_1, \boldsymbol{\eta}, \kappa, \delta, \Psi) P(\Sigma_{22}, \Sigma_{21} | \Sigma_{11} \delta, \Psi) = \\
&= \mathcal{N}(\boldsymbol{\mu}_1 | \boldsymbol{\eta}_1, \frac{1}{\kappa}\Sigma_{11}) \mathcal{IW}(\Sigma_{11} | \delta, \Psi_{11}) \qquad \text{(A.8)}
\end{aligned}$$

$\square$