Context-specific Independence Mixture Models for Cluster Analysis of Biological Data

Benjamin Georgi

März 2009

Dissertation zur Erlangung des Grades
eines Doktors der Naturwissenschaften (Dr. rer. nat.)
am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

Gutachter:
Prof. Dr. Martin Vingron
Prof. Dr. Jörg Schultz

1. Referent: Prof. Dr. Martin Vingron
2. Referent: Prof. Dr. Jörg Schultz
Tag der Promotion: 10.6.2009

Contents

List of Figures

List of Tables

Preface

Acknowledgments

First of all, I would like to thank my supervisor Alexander Schliep. Without his ongoing support, guidance and encouragement this work would not have been possible.

I would like to thank Jörg Schultz for the fruitful collaboration on protein subfamilies. Also, my thanks go to Silke Sperling for making available the heart disease data and discussions of the results. I am grateful to Julia Lassere for discussion and excellent feedback on the manuscript. This work was conducted in the department of Bioinformatics at the Max-Planck-Institute for Molecular Genetics. I would like to thank all past and present colleagues for many interesting discussions and for being good company. In particular, I would like to thank Martin Vingron for his supervision and support of this work. Finally, my thanks go out to my family and friends.

Publications

Several parts of this thesis are based on prior publications. Aspects of the method developed in chapter 3 were presented in a paper at the *Annual Conference of the German Classification Society* 2007 [68]. The study on transcription factor binding sites in chapter 5 was presented on the *International Conference on Intelligent Systems in Molecular Biology* 2006 and published in Bioinformatics [65]. The study of protein subfamilies in chapter 6 contains results from a paper published at the *European Conference on Machine Learning* 2007 [67] and at the workshop on *Data Mining in Functional Genomics and Proteomics* at the same conference [66]. In addition some collaborative work during the course of my Phd studies resulted in an additional publication not described in the thesis [75].

Chapter 1

Introduction

This thesis is concerned with the analysis of biological data by statistical clustering approaches. In this chapter we give an introduction into the general properties of modern biological data sets, the challenges they pose for data analysis and describe in some more detail the specific data sets we are concerned with. Finally, the content and main scientific contributions of this thesis will be summarized.

1.1 Biological Mass data

The advent of modern, high-throughput experimental techniques has lead to new wealth of data for all aspects of molecular biology. While the nature of these data sets is extremely diverse, there are certain properties which are shared by many of these types of data.

- The data is high-dimensional and only a subset of the features can be expected to be informative for the purpose of an analysis.
- The values of the data points are distorted by noise and the data set contains a non-negligible number of missing values. Also, many biological data sets will include outliers due to experimental artifacts.
- The data set incorporates multiple sources of data from different domains (e.g different experimental methods, geno- and phenotypic data, etc.), where the relative relevance for the biological question to be addressed, as well as potential dependencies between the different sources are unknown.

These properties make the analysis of such data a challenging task. The presence of many uninformative features increases the difficulty of picking up on the regularities of interest, as trends in the data are overshadowed by the cumulative effect of the uninformative features. The presence of noise as well as missing values require to be dealt with in a principled manner. Finally, the possibility of integrating several, heterogeneous sources of data in a single analysis is of increasing importance.

In the following sections we are going to give short, fairly general introductions into the types of biological data this work is concerned with. For more details, refer to any mo-

lecular biology textbook (e.g [181]). More detailed information on the specific biological backgrounds of the data sets we analyzed will be given in the respective chapters.

Genomic Sequences

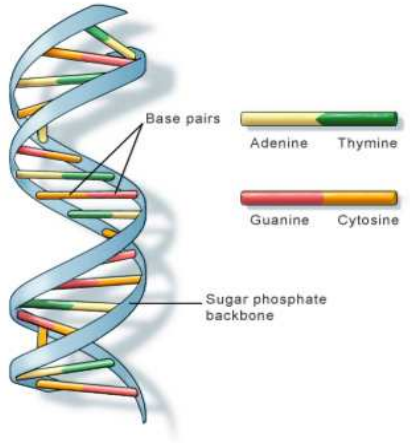


Figure 1.1: *DNA double helix. The helix is formed by the complementary pairing of A–T and G–C pairs of nucleotides. (Image courtesy of the US National Library of Medicine)*

The building plan of any living organism, its genetic information, is stored and passed on in form of deoxyribonucleic acid (DNA) molecules. In the cell the DNA is organized as two polynucleotide chains, or strands, which are intertwined in the famous double helix structure (shown in Fig. 1.1). In genomic DNA there are four nucleotides, or bases, which form the consistent parts of the DNA chain. These four bases are adenine (A), cytosine (C), guanine (G),and thymine (T). The double helix is formed by the complementary pairing of the two strands by hydrogen bonds of A–C and G–T pairs of nucleotides.

Much of the progress in modern genetics has been driven by the progress of DNA sequencing technologies and the ever increasing number of completely sequenced genomes. The availability of such genomic sequences allows the study of questions such as the prediction of gene positions (*gene prediction* [92, 94]), evolutionary relatedness of different organisms and species (*phylogenetics* [27]) and the common genetic basis of closely and distantly related organisms (*comparative genomics* [80]).

Protein Sequences

Whereas DNA is the medium in which genetic information is stored in the living cell, proteins are how this information is expressed. Proteins fulfill specific functions in the cell and the interplay of many (i.e. on the order of hundreds of thousand for humans) proteins form the whole organism. On the DNA level information how to build each protein is stored in specific stretches of the nucleotide chain, the genes.

The information in the genes is turned into proteins by the two processes *transcription* and *translation*. In the *transcription* step, the sequence of a gene is copied into the *messenger* ribonucleic acid (mRNA). The mRNA is needed to carry the genetic information out of the nucleus, where the DNA resides, into the cell plasma. There, the mRNA is read by molecular factories, the ribosomes, which actually build the proteins. Each protein is a polypeptide chain of twenty different amino acids (see appendix C). The order of amino acids in a protein is determined by the order of nucleotide in the respective gene (via the mRNA) according to the *genetic code*.

In vivo, proteins exist in a three dimensional fold of the linear amino acid sequence, the protein structure. This structure is typically considered on three levels. The first and simplest, the *primary* structure, is simply the amino acid sequence itself. The *secondary* structure is the folding of the amino acid chain into typical local structures, the α–helices and β–sheets, which are connected by loops. Finally, the *tertiary* structure is the complete three dimensional fold of the whole chain. Fig. 1.2 shows an example protein structure. The secondary structure elements are shown in red (α–helices) and yellow (β–sheets). Loop regions are depicted green. The fold each protein takes, determines which amino acids, in which configurations, are presented to the outside medium and thereby also which function it performs in the organism. In this work we focus on the analysis of *primary* structure data, i.e. amino acid sequences.

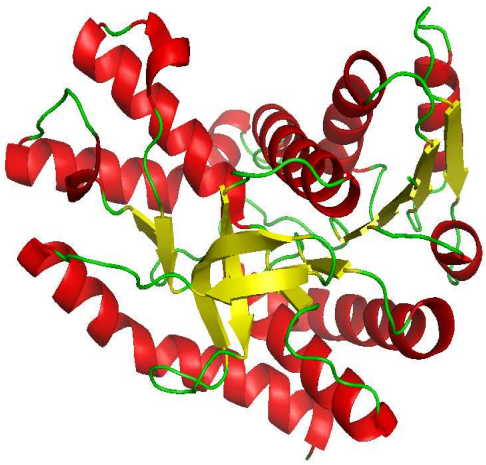


Figure 1.2: *Example protein structure. Secondary structure elements are depicted in red (α–helices), yellow (β–sheets) and green (loops).*

By analyzing protein sequences one can study questions such as the prediction of protein function based on the similarity of the amino acid sequences (*protein homology*, e.g. [182]), discovery of subsequences with a specific function which occur in many proteins (*protein domain discovery*, e.g. [34, 35]) and prediction of the three dimensional fold of a protein (*structure prediction*, e.g. [23, 134]).

Complex Disease Phenotypes

A different kind of data set arises from the study of complex genetic diseases. The analysis of genetic diseases has classically been directed towards establishing direct links between cause, a genetic variation, and effect, the observable deviation of phenotype. For complex diseases which are caused by multiple factors and which show a wide spread of variations in the phenotypes this is unlikely to succeed. Recently, vast efforts are being undertaken to collect data sets which allow the study and, potentially, elucidation of the various genetic factors contributing to a diseases' mode of inheritance and course in an individual (e.g. [69, 83]). Often these data bases contain both genotypic and phenotypic information. The genotypic features of such a data set either are given by genomic regions which have been linked to the disease in prior studies or, increasingly often, by whole–genome genotyping based on next generation sequencing techniques. The phenotypes consist of clinical features relevant to the disease. These can be as diverse as questionnaire scores for psychological disorders and morphological abnormalities for physically manifested diseases. In this work we focus on the analysis of disease phenotypes. Due to the high variability of complex phenotypes a structuring of distinctive phenotype patterns is often a crucial first step in the analysis.

Important questions with regards to the study of such diseases are for instance the identification of candidate genes and genomic regions using linkage analysis (e.g. [73, 100]) or candidate gene approaches [181].

1.2 Thesis Overview

The main focus of this thesis is the detection of meaningful subgroups in biological data sets where noise and the presence of uninformative features confound the regularities given by biological subgroupings. That is, we are concerned with clustering and cluster analysis of biological data. Clustering techniques attempt to find subgroups of similar samples in the data. It is both exploratory and *unsupervised*, i.e. the biological interpretation of the discovered groups is not necessarily clear and assignments of samples to known categories are not given *a priori*.

A classical statistical framework for performing clustering are mixture models (see chapter 2). Mixture models have attractive properties for analyzing biological data. Namely that due to their probabilistic nature, mixtures acknowledge the inherent ambiguity of any group assignment in exploratory biological data analysis, in a structured and theoretically sound way. In chapter 2 we also describe the parameter learning by expectation maximization (EM) and give a practical introduction for using mixtures for clustering. The chapter is concluded by a description of the Bayesian formulation of mixture models. Chapter 2 is a review of established research on conventional mixture models and lays the foundation of the extensions described in chapter 3.

In chapter 3 the *context-specific independence* (CSI) extension to the mixture framework is introduced and we give a novel formulation of CSI in mixture models which conveys additional attractive properties for practical data analysis. This is followed by an update of the Bayesian formulation for CSI mixtures. Also, we describe the structural EM algorithm necessary to learn CSI mixtures from data and derive the parameter estimators. In the last section of the chapter we discuss some practical advantages of CSI mixtures for cluster analysis. Chapter 3 draws from prior work on CSI mixtures which employed a less rich CSI formulation and updates the established structural EM algorithm for this new CSI formulation.

Chapter 4 deals with practical aspects of learning CSI mixtures from data. The complexity of the structure learning problem is discussed and various strategies for reducing the complexity in practice are introduced and evaluated. This chapter also gives results on an approach for reducing the running time of the structure learning.

In this thesis CSI mixture based clustering was applied to three biological applications. In chapter 5 we present the first application of CSI mixtures for the modeling of transcription factor binding sites (TFBS). We show that CSI is more suited to the problem than the conventional mixtures previously applied and examine the biological implications of the subgroups found.

In chapter 6 we describe the application of CSI mixture for clustering of protein subfamilies with simultaneous prediction of functional residues. We also examine some challenges posed by protein sequence data and present a model extension in form of a novel Dirichlet mixture prior to address them.

The third application deals with the clustering of heart disease phenotypes (chapter 7). The aim of this analysis being to detect groups of patients which are characterized by different phenotype patterns. These groups then also might share some causal variant on the genomic level. That means in this setup the clustering amounts to detection of disease subgroups.

Finally, in chapter 8 the results and implications of this work will be discussed.

Chapter 2

Finite Mixture Models

Mixture models are a powerful and versatile class of probabilistic models for density estimation and data analysis. The central paradigm of the mixture framework is that the observed data is generated by a number of different and unobservable underlying processes. Each of these processes is represented by a distribution (or density in case of continuous data) and the combination of these *component* distributions by a convex combination then forms the mixture distribution. Mixtures are not only theoretically capable of representing arbitrary distributions [129], in practice they are also an efficient alternative for more complex models such as Bayesian networks [117]. One of the first papers introducing mixtures was the 1898 Pearson paper [141] which dealt with the modeling of the size distribution of a heterogeneous population of crabs. Since then mixture models have been applied in numerous fields and settings, including sociology (in form of *latent class* models [106]) or as building blocks of neural networks [19].

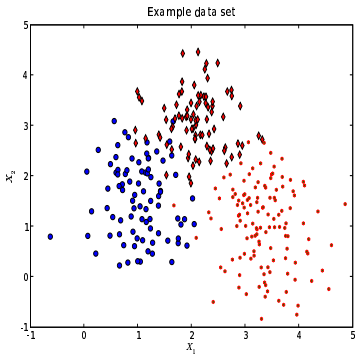
In this chapter we are going to introduce finite mixture models, the parameter learning algorithm, describe how mixtures can be used for clustering and the extension of mixtures to the Bayesian framework.

2.1 Mixture Models

Let $X = X_1, ..., X_p$ denote random variables (RVs) representing the features of a p dimensional data set D with N samples $x_i, i = 1, ..., N$ where each x_i consists of a realization $(x_{i1}, ..., x_{ip})$ of $(X_1, ..., X_p)$. A K component mixture distribution is given by

$$P(x_i|\Theta) = \sum_{k=1}^{K} \pi_k P(x_i|\theta_k), \tag{2.1}$$

where the $\pi_k \geq 0$ are the mixture coefficients with $\sum_{k=1}^{K} \pi_k = 1$. For our purpose each component distribution $P(x_i|\theta_k)$ is defined as a product distribution over $X_1, ..., X_p$ pa-

a) 

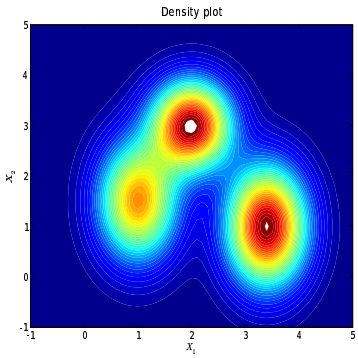
b) 

Figure 2.1: ***a)*** *Example data set from a Gaussian mixture with three components over two-dimensional data (X_1, X_2).* ***b)*** *Corresponding mixture density heat map with three normal components.*

rameterized by parameters $\theta_k = (\theta_{k1}, ..., \theta_{kp})$,

$$P(x_i|\theta_k) = \prod_{j=1}^{p} P(x_{ij}|\theta_{kj}). \tag{2.2}$$

Then each of the $P(x_{ij}|\theta_{kj})$ is a distribution over a X_j, conditional on a mixture component k. The form the parameters θ_{kj} take depends how feature X_j is distributed (see section 2.1.1 for examples). We denote the collection of all θ_{kj} and the weight vector $\pi = (\pi_1, ..., \pi_K)$ as $\Theta = (\pi, \theta_1, ..., \theta_k)$. Then Θ completely parameterizes the mixture.

Fig. 2.1 shows an example data set and corresponding mixture density for two continuous, normally distributed features and a three component mixture. Fig. 2.1a) shows a plot of an example data set where different colors denote samples arising from different components. Fig. 2.1b) shows a heat map of the mixture density the data was generated with. Each of the three normal components can be seen as modes of the density and it can be seen that the components overlap.

The likelihood $P(D|\Theta)$ for data set D with N samples is simply the product over the mixture density at each sample

$$P(D|\Theta) = \prod_{i=1}^{N} P(x_i|\Theta). \tag{2.3}$$

In order to illuminate the model properties it is instructive to examine Θ more closely. In addition to the mixture weights π, the model parameterization includes one set of parameters θ_{kj} for each component in the model and feature X_j in the data. This can be visualized as shown in Fig. 2.2a). The example shows the parameter matrix for a mixture with 5 components $C_1, ..., C_5$ and 4 features $X_1, ..., X_4$. It can be seen that the model parameters are arranged in a matrix spanned by the mixture components and the features of the data set. A

more abstract representation of the same model can be obtained by omitting the parameter variable names from the parameterization matrix. This yields the *model structure matrix* (Fig. 2.2b)). For conventional mixture models, the model structure does not look particularly interesting as every component has a separate set of parameters θ_{kj} for each feature. This will change with the introduction of context-specific independence in chapter 3.

a)

		X_1	X_2	X_3	X_4
C_1	π_1	θ_{11}	θ_{12}	θ_{13}	θ_{14}
C_2	π_1	θ_{21}	θ_{22}	θ_{23}	θ_{24}
C_3	π_1	θ_{31}	θ_{32}	θ_{33}	θ_{34}
C_4	π_1	θ_{41}	θ_{42}	θ_{43}	θ_{44}
C_5	π_1	θ_{51}	θ_{52}	θ_{53}	θ_{54}

b)

	X_1	X_2	X_3	X_4
C_1				
C_2				
C_3				
C_4				
C_5				

Figure 2.2: ***a)*** *Model parameterization matrix for a five component mixture over four features.* ***b)*** *Corresponding model structure matrix.*

The assumption of independence between elements of X which yields the convenient decomposition in Eq. (2.2) is a rather strong one and bears further discussion. The component product distributions $P(x_i|\theta_k)$ are also known as naïve Bayes models. These models have been successfully used on a large variety of applications on areas as diverse as emotion recognition [133], credit scoring [40], diagnosis of acute abdominal pain [42] or text mining [147, 162]). It has been found that despite its simplicity naïve Bayes performs surprisingly well for a broad range of applications. This is true even in situations where the independence assumption is not necessarily met by the data [58, 76, 137, 143, 154, 186]. Also, naïve Bayes is a competitive and efficient alternative to Bayesian networks for general density estimation [117]. Finally, the optimality of the naïve Bayes classifier has been shown for specific problem settings [47, 98]. That being the case, it should be stressed that there has been considerable progress for classification problems and state-of-the-art methods such as *support vector machines* [36] can be expected to outperform naïve Bayes. However, since this work is primarily concerned with clustering problems, this has no direct relevance.

It is important to realize that the independence assumptions for X are conditional on the mixture components k. In other words, the assumption is that the strongest dependencies between features are captured by the K mixture components and that, *given* a specific component the features can be treated as independent. This kind of independence assumption, while still a simplification, has proved to be very useful in many applications.

One advantage of adopting naïve Bayes models as component distributions is that it conveys great flexibility in modeling different distributions in X. For instance continuous and discrete RVs can be seamlessly integrated into the same model. This flexibility has been extensively made use of by researchers to design mixture models for a wide variety of applications, including distributions such as multinomial [65, 67, 153], Gaussians [74, 136, 138, 149], exponential [11], Poisson [108], uniform [43] and Dirichlet [170]. In this work we are going to focus on the multinomial and normal distributions due to the nature of the data under consideration. It should be noted however, that all that is required

to build mixtures for any distribution from the exponential family [5], is to plug in the appropriate density functions and parameter estimators into the framework we are about to describe.

2.1.1 Atomic Distributions

The K distributions $P(x_{ij}|\theta_{kj})$ over feature X_j in a mixture with naïve Bayes components can be specified freely from the exponential family to match the data domain of feature X_j. It should be noted however, that these distributions are only *atomic* in the sense that each one models feature a single feature X_j. Each of the X_j could be vector–valued in itself and conceptually each of the θ_{kj} can be multivariate or even a mixture distribution over X_j in itself. In other words it is possible to have both univariate and multivariate distributions as parts of a naïve Bayes component model.

In the following we focus on normally distributed and discrete valued data. For the Gaussian data we have $\theta_{kj} = (\mu_{kj}, \sigma^2_{kj})$ where μ_{kj} and σ^2_{kj} parameterize the Gaussian density function

$$P(x_{ij}|\theta_{kj}) = \frac{1}{\sigma_{kj}\sqrt{2\pi}} \exp\left(-\frac{(x_{ij}-\mu_{kj})^2}{2\sigma^2_{kj}}\right). \tag{2.4}$$

If all X_j are distributed as Gaussians, the product over univariate Gaussians in the component distribution $P(x_i|\theta_k)$ is equivalent to a multivariate Gaussian with diagonal covariance matrix. The assumption of diagonality for the covariance is often made to avoid serious numerical problems with estimating full covariance matrices on limited amounts of data [127]. The Gaussian distribution is often used to model continuous measurements of biological quantities. One important example for such quantities are the various experimental techniques for measuring gene expression [46, 140, 160].

In the case of discrete data we have $\theta_{kj} = (\phi_{kj})$, where $\phi_{kj} = (\phi_{kj1}, ..., \phi_{kjM})$ is a stochastic vector of length M defining a distribution over an alphabet Σ with M symbols. Then ϕ_{kj} parameterizes the discrete probability mass function given simply by the element of ϕ_{kj} corresponding to the symbol x_{ij} i.e.

$$P(x_{ij}|\theta_{kj}) = P(x_{ij} = \Sigma_s|\phi_{kj}) = \phi_{kjs}, \tag{2.5}$$

where Σ_s denotes the s'th symbol in the alphabet.

In bioinformatics the most important source of discrete data is inarguably the vast body of known biological sequences. This includes DNA and protein sequences. As mentioned earlier, later chapters will give example applications for DNA and protein data (chapters 5 and 6).

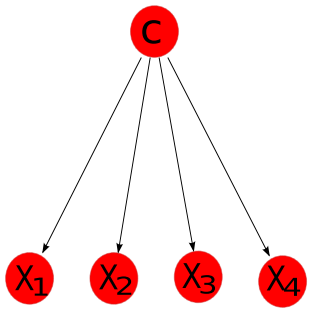
2.1.2 Mixture Models from Different Perspectives

In mathematics the same or similar concepts often arise in different subfields in a process reminiscent of convergent evolution. Considering how mixture models are expressed from the perspective of different subfields, is instructive to gain deeper understanding of the constraints and flexibilities inherent to the model formulation used.

Mixtures as Bayesian Networks

A Bayesian network (BN) [81] defines the joint distribution of a number of RVs X_j by encoding the conditional independence between RVs in a directed acyclic graph. The central assumption of the BN formalism is that RVs are only dependent on their parents in the graph. This allows for factorization of the joint likelihood and efficient inference. As described in section 2.1, in the mixture framework all RVs X_j are conditionally independent given the component. In order to represent the example given in Fig. 2.2b) as a BN, we introduce the component indicator variable C. C is a discrete RV which takes values in the set of component indices $1, ..., K$. The resulting BN graph structure is shown in Fig. 2.3a). The conditional distributions with arise from the conditional independence statements encoded in the graph structure are usually expressed in so called *conditional probability tables* (CPTs). Fig. 2.3b) shows the CPT for feature X_2 for the example graph in a). Since X_2 is only dependent on C, there exists a separate conditional distribution $\theta_{k2}, k = (1, ..., 5)$ for each possible value of C.

a)



b)

C	X_2
1	θ_{12}
2	θ_{22}
3	θ_{32}
4	θ_{42}
5	θ_{52}

Figure 2.3: *a) Bayesian network graph for a mixture distribution with four features. $X_1 - X_4$ are conditionally independent given the component indicator C. b) CPT for feature X_2. The conditional distributions of X_2 given the value of C are listed.*

Mixtures as Hidden Markov Models

Hidden Markov models (HMMs) are time discrete stochastic processes which have been used extensively for such applications as analysis of time courses [150, 161] and biological sequences [94, 97]. An HMM consists of a number of hidden states, each of which has an emission distribution, describing the observed data and a transition distribution which describes the dynamics within the state space. For more details on HMMs refer to [150].

The HMM topology which defines a model equivalent to the five component mixture introduced in Fig. 2.2b) is shown in Fig. 2.4. There, each of the five linear chains of four states corresponds to a mixture component over four features. The transitions distribution from the START state to any of the chains is just the mixture weights π. From the HMM

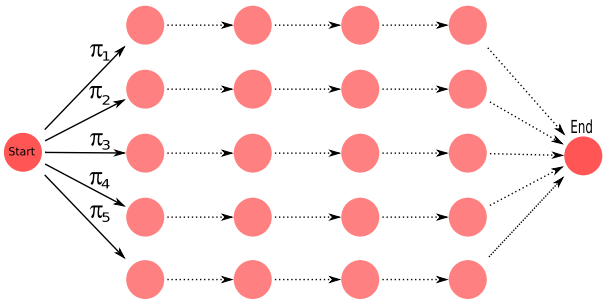


Figure 2.4: *HMM topology equivalent to the five component mixture from Fig. 2.2b) The dashed arrows denote transitions with probability one.*

perspective, a mixture is an HMM with K parallel linear state paths and a fixed observation length p. This relation between mixtures and HMMs means that it is straightforward to adopt HMMs as component distributions of a mixture [160].

2.1.3 Sampling from a Mixture

Since mixtures are *generative* models, it is straightforward to sample observations from a given model. For each sample $x = (x_1, ..., x_p)$ first a component $k \in (1, .., K)$ is chosen by sampling from the mixture weight distribution π, i.e.

$$k \sim \pi.$$

In the next step the elements of x are sampled from the component distribution θ_k,

$$x \sim \theta_k$$

s.t. $x_j \sim \theta_{kj}$ for each j = $1, ..., p$. The straightforward generation of artificial data from a given mixture is very useful for tasks such as the validation of the parameter estimation procedures and assessment of clustering performance. One typical setup for the latter would be to sample a data set from a given mixture while recording the true component labels of each sample. Then, a new model is learned from that data set and the clustering performance can be assessed by comparing to the true labels.

2.2 Expectation Maximization (EM) Algorithm

The central learning task that needs to be addressed for a data set D is inferring the values of the parameters Θ. The reason that one cannot straightforwardly calculate maximum

likelihood (ML) estimates for Θ is that the assignment of samples to components in the mixture is unknown. This is referred to as an *unsupervised* learning problem. Classically, this situation is also often referred to as an incomplete data problem in which the observed data D is joined by the unknown component assignments H to form the complete data $D_c = (D, H)$. In the case of complete data (i.e. the *supervised* case), obtaining maximum likelihood estimates $\hat{\Theta}$ for the parameters of a mixture is straightforward. In the incomplete data case however, $\hat{\Theta}$ cannot be calculated analytically. The standard technique to arrive at parameter estimates $\hat{\Theta}$ in the incomplete data case is the *Expectation Maximization* (EM) algorithm [44].

2.2.1 General Formulation

The principle idea of the EM algorithm is to replace the unknown, hidden values with their conditional expectations based on the current parameters and the data. Once these expectations have been computed, new parameters $\hat{\Theta}$ can be analytically computed by substituting the hidden values by their conditional expectations and treating the problem as the complete data case. Iterations of these two steps will converge to a local maximum of the likelihood function [44].

Formally, the aim is to find the parameters $\hat{\Theta}$ which maximize the probability of the observed data $P(D|\ \Theta)$, i.e. the ML estimates

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}}\, P(D|\ \Theta).$$

In order to achieve this, an auxiliary Q function is defined. The Q function is the conditional expectation of the likelihood of the complete data $D_c = (D, H)$ given the observed data D and a parameterization Θ^{t-1}. This yields the Q function as

$$Q(\Theta, \Theta^{t-1}) = E[\log\ P(D, H|\Theta)|D, \Theta^{t-1}], \tag{2.6}$$

where the observed data D and the current model parameters Θ^{t-1} can be considered constant. The missing data H is unknown. Finally, the new parameters Θ are the target of the maximization. The Q function can be rewritten by summing over the unknown hidden data $h \in H$ (assuming the hidden data H is discrete, otherwise integration over H is required).

$$E[\log P(D, H|\Theta)|D, \Theta^{t-1}] = \sum_{h \in H} \log P(D, h|\Theta) P(h|D, \Theta^{t-1})\ dh \tag{2.7}$$

where $P(h|D, \Theta^{t-1})$ is the distribution of the hidden values H conditioned on the current parameters Θ^{t-1} and the data D. By integrating over the hidden values H, the Q function becomes a deterministic function in Θ, which can be maximized analytically for distributions from the exponential family.

In the following we are going to show one of the central results of EM algorithm theory [44]. Namely that that maximizing Eq. (2.6) with respect to Θ, i.e setting

$$\Theta^t = \underset{\Theta}{\operatorname{argmax}}\, Q(\Theta, \Theta^{t-1}) \tag{2.8}$$

also increases the likelihood $P(D|\Theta)$, i.e.

$$\log P(D|\Theta) \geq \log P(D|\Theta^{t-1}).$$

This can be shown as follows: From the $\log$-ratio of the two likelihoods

$$\log P(D|\Theta) = \log P(D, H|\Theta) - \log P(H|D, \Theta)$$

the following is obtained by taking the conditional expectation of H with respect to D and the current parameters Θ

$$\log P(D|\Theta) = \sum_{h \in H} \log P(D, h|\Theta) P(h|D, \Theta^{t-1}) - \sum_{h \in H} \log P(h|D, \Theta) P(h|D, \Theta^{t-1}). \tag{2.9}$$

In terms of the definition in Eq. (2.6) this can be rephrased as

$$\log P(D|\Theta) = Q(\Theta, \Theta^{t-1}) - \sum_{h \in H} \log P(h|D, \Theta) P(h|D, \Theta^{t-1}). \tag{2.10}$$

By applying the same transformation to $\log P(D|\Theta^{t-1})$ it follows that

$$\begin{aligned} \log P(D|\Theta) - \log P(D|\Theta^{t-1}) = Q(\Theta, \Theta^{t-1}) - Q(\Theta^{t-1}, \Theta^{t-1}) \\ + \sum_{H \in H} P(h|D, \Theta^{t-1}) \log \frac{P(h|D, \Theta^{t-1})}{P(h|D, \Theta)} \end{aligned} \tag{2.11}$$

The terms in the sum just form the relative entropy (also known as the Kullback-Leibler divergence) [96] between the two distributions $P(H|D, \Theta^{t-1})$ and $P(H|D, \Theta)$. Since the relative entropy is always nonnegative, it follows that

$$\log P(D|\Theta) - \log P(D|\Theta^{t-1}) \geq Q(\Theta, \Theta^{t-1}) - Q(\Theta^{t-1}, \Theta^{t-1}). \tag{2.12}$$

Substituting the optimal parameters Θ^t from Eq. (2.8) into Eq. (2.12) yields

$$\begin{aligned} \log P(D|\Theta^t) - \log P(D|\Theta^{t-1}) &\geq Q(\Theta^t, \Theta^{t-1}) - Q(\Theta^{t-1}, \Theta^{t-1}) \\ &\geq Q(\Theta, \Theta^{t-1}) - Q(\Theta^{t-1}, \Theta^{t-1}) \\ &\geq 0 \end{aligned}$$

which also implies

$$\log P(D|\Theta^t) > \log P(D|\Theta^{t-1}).$$

This means that by maximizing the conditional expectation of the full likelihood (Eq. (2.6)) given some arbitrary parameters Θ^{t-1} and the data with respect to new parameters Θ^t, we can obtain an improved likelihood $\log P(D|\Theta^t)$. This immediately suggests an iterative procedure in which Θ^t from the previous step becomes Θ^{t-1} for the next. It can be shown that these iterations converge to a local optimum of the likelihood function [44].

Note that the condition of Θ^t maximizing $Q(\Theta^t, \Theta)$ in Eq. (2.8) can be relaxed to requiring Θ^t only to increase $Q(\Theta^t, \Theta)$. This is referred to as *generalized* EM algorithm (e.g. [45, 52, 146]) and is useful in situations were maximization of Q if difficult.

In summary, the EM procedure for finding ML estimators for Θ consists of iterations over two steps:
EM Algorithm:

1. **Expectation Step**: Evaluate $Q(\Theta^t, \Theta^{t-1})$ by substituting the conditional expectations of H.
2. **Maximization Step**: Maximize Q with respect to Θ^t, i.e. $\Theta^t = \text{argmax}_\Theta\, Q(\Theta, \Theta^{t-1})$

2.2.2 EM for Mixture Models

In the following we are going to derive the EM for mixture models based on the general formulation of in the previous section. Given the mixture

$$P(x_i|\Theta) = \sum_{k=1}^{K} \pi_k P(x_i|\theta_k) \tag{2.13}$$

the log–likelihood function for a data set $D = x_1, ..., x_N$ is

$$\log L(\Theta|D) = \sum_{i=1}^{N} \log \left[\sum_{k=1}^{K} \pi_k \; P(x_i|\theta_k) \right], \tag{2.14}$$

which, as mentioned previously, cannot be maximized directly due to the sum within the logarithm. To get around this issue we consider the hidden data H, which is given by the assignments of samples $x_1, ..., x_N$ to components $(1, ..., K)$. One way to formalize this is to define the space of hidden data H as the set of $K \times N$ binary matrices with exactly one element equal 1 in each column. For one configuration of the hidden data $h \in H$, a value of $h_{ki} = 1$ then indicates that x_i was generated by component k. Then the joint distribution of the observed and hidden data is given by the complete data log–likelihood

$$\log L(\Theta|D, H) = P(D, H|\Theta) = \sum_{k=1}^{K} \sum_{i=1}^{N} h_{ki}(\log \pi_k + \log P(x_i|\theta_k)), \tag{2.15}$$

i.e. simply the sum over the likelihoods of samples x_i, where h_{ki} indicates which component contributed each sample. Again the EM Q function is

$$Q(\Theta, \Theta^{t-1}) = E[\log P(D, H|\Theta)|D, \Theta^{t-1}] = \sum_{h \in H} \log P(D, h|\Theta) P(h|D, \Theta^{t-1}) \quad (2.16)$$

where h is a possible configuration of the component assignment indicator matrix H, $P(D, H|\Theta)$ is the complete data distribution and

$$P(H|\, D, \Theta^{t-1}) = \prod_{i=1}^{N} \prod_{k=1}^{K} P(h_{ki} = 1|x_i, \Theta^{t-1})^{h_{ki}}$$

is the distribution of the hidden data given D and Θ^{t-1}. The terms in the product are the posteriors of component membership for each sample. By applying Bayes' rule, this posterior is given by

$$\begin{aligned} \tau_{ki} = P(h_{ki} = 1|D, \Theta^{t-1}) &= \frac{P(h_{ki} = 1) P(x_i|h_{ki} = 1, \Theta^{t-1})}{P(x_i|\Theta^{t-1})} \\ &= \frac{\pi_k P(x_i|\theta_k)}{\sum_{k=1}^{K} \pi_k P(x_i|\theta_k)}. \end{aligned} \quad (2.17)$$

This posterior is crucial for both the parameter estimation in the EM framework as well as using mixture models for clustering (see section 2.3). $P(h_{ki} = 1|D, \Theta^{t-1})$ gives the probability that a sample x_i was generated by component k. For ease of notation in the following we will refer to $P(h_{ki} = 1|D, \Theta^{t-1})$ as τ_{ki}.

2.2.3 Parameter Estimators

The EM objective function for the mixture case Eq. (2.16) can be formulated as

$$Q(\Theta, \Theta^{i-1}) = \sum_{k=1}^{K} \sum_{i=1}^{N} \tau_{ki} \left(\log \pi_k + \log P(x_i|\theta_k)\right) \quad (2.18)$$

by expanding the sum over $h \in H$ and rearranging terms [17, 129].

Based on Eq. (2.18) ML estimators for the parameters in Θ can be derived by analytical maximization of the Q function under appropriate side constraints with respect to the model parameters in Θ^t [17].

As an example we give details for the derivation of the estimators for π. First note that

Eq. (2.18) can be written as

$$Q(\Theta, \Theta^{i-1}) = \sum_{k=1}^{K}\sum_{i=1}^{N} \tau_{ki} \log \pi_k + \sum_{k=1}^{K}\sum_{i=1}^{N} \tau_{ki} \log P(x_i|\theta_k), \tag{2.19}$$

and that there are no terms containing both π_k and θ_k. Therefore π_k and θ_k can be maximized separately. In order to ensure stochasticity we introduce the side condition $\sum_{k=1}^{K} \pi_k = 1$ into the first sum on the right-hand side of Eq. (2.19) which leads to partial derivatives of the Lagrangian (e.g. [101]) with respect to $\pi_k (k = 1, ..., K)$ and λ

$$\frac{\delta Q}{\delta \pi_k}\left[\sum_{k=1}^{K}\sum_{i=1}^{N} \tau_{ki} \log \pi_k + \lambda(\sum_{k=1}^{K} \pi_k - 1)\right] =$$

$$\sum_{i=1}^{N} \frac{1}{\pi_k}\tau_{ki} + \lambda = 0, \tag{2.20}$$

$$\frac{\delta Q}{\delta \lambda} = \left[\sum_{k=1}^{K} \pi_k - 1\right] = 0. \tag{2.21}$$

Solving Eq. (2.20) for π_k and substituting into Eq. (2.21) yields $\lambda = -N$. Resubstitution into Eq. (2.20) yields the estimators for the mixture weights π in each time step as

$$\hat{\pi}_k = \frac{\sum_{i=1}^{N} \tau_{ki}}{N} \quad (k = 1, ..., K). \tag{2.22}$$

Estimation of the component parameters θ_k requires taking derivatives with respect to θ_k for the second sum in Eq. (2.19). For the naïve Bayes component distributions this sum further simplifies to

$$\sum_{k=1}^{K}\sum_{i=1}^{N} \tau_{ki} \log P(x_{i1}|\theta_{k1}) + ... + \tau_{ki} \log P(x_{ip}|\theta_{kp}),$$

which means that the derivatives for the individual θ_{kj} also can be taken separately. For atomic distributions θ_{kj} from the exponential family there are closed form solutions for the ML estimators. For the univariate Gaussian distribution $\theta_{kj} = (\mu_{kj}, \sigma^2_{kj})$ the ML estimators for parameters μ_{kj} and variance σ^2_{kj} [129] are

$$\hat{\mu}_{kj} = \frac{\sum_{i=1}^{N} \tau_{ki} x_{ij}}{\sum_{i=1}^{N} \tau_{ki}} \tag{2.23}$$

and

$$\hat{\sigma}^2_{kj} = \frac{\sum_{i=1}^{N} \tau_{ki}(x_{ij} - \hat{\mu_{kj}})^2}{\sum_{i=1}^{N} \tau_{ki}}. \tag{2.24}$$

In case of discrete distributions $\theta_{kj} = \phi_{kj}$ estimates for parameters $\phi_{kj} = (\phi_{kj1}, ..., \phi_{kjM})$ over some M–symbol alphabet Σ are given by

$$\hat{\phi}_{kjs} = \frac{\sum_{i=1,\ x_{ij}=\Sigma_s}^{N} \tau_{ki}}{\sum_{i=1}^{N} \tau_{ki}} \quad (s = 1, ..., M). \tag{2.25}$$

2.2.4 Drawbacks of the EM Algorithm

While the EM algorithm allows efficient parameter estimation, the algorithm also has a number of drawbacks. The first and foremost concern being that convergence is only guaranteed locally. The quality of such a local maximum relative to the global maximum can still be arbitrarily poor. The standard approach to address this issue is to run the EM procedure many times from different initial parameter sets, thereby exploring the likelihood surface and in the end retaining the best parameters found. An simplified example of such a likelihood surface is shown in Fig. 2.5a). The x-axis represents the model parameters Θ, the y-axis shows the corresponding likelihood function. From different starting values Θ^0, the EM procedure converges (dashed arrows) to different local maxima shown in red and blue. Therefore, some care has to be taken with the choice the different initial parameterizations Θ^0. One possible approach which works well in practice is to randomly assign samples to components and then perform an M-Step update to obtain Θ^0.

Another issue is EM's sensitivity to outliers in the data. A small number of uncharacteristic data points can cause the EM procedure to get trapped in spurious local maxima at the edges of the parameter space. One example of that would be a Gaussian mixture density containing one or several components with very small variance σ^2. An example of such a density is shown in Fig. 2.5b). The peaked component to the right, contributes disproportionately to the whole likelihood by overfitting a few outlying data points.

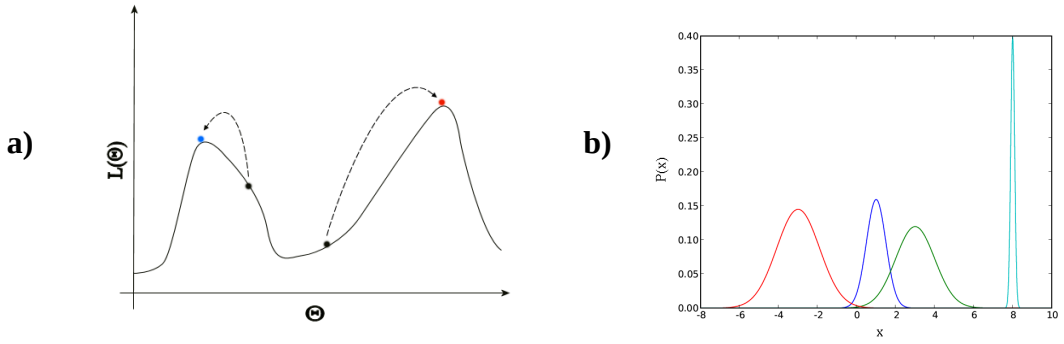


Figure 2.5: *a) Simplified likelihood surface with two maxima. Depending on the initial parameters Θ^0 the EM procedure converges to a different maximum. b) Example of a Gaussian mixture which attains a spurious maximum by overfitting an outlier.*

This problem can be addressed by adding a dedicated noise component to reduce the impact

of outliers (see section 2.3.4) or by a regularization of the parameter estimates in a Bayesian setting (see section 2.4).

2.3 Mixture Models for Clustering

One major application of mixture models is clustering. Clustering gives a decomposition of the N samples of a given data set into K subgroups. Among the classical approaches to clustering are the k-means algorithm [119], self-organizing maps [156] and hierarchical clustering approaches [99]. One important feature of the mixture framework is that due to its probabilistic nature, it naturally represents overlapping clusters. In the mixture framework, each cluster is identified with one of the K components and the components parameters θ_k capture the regularities which characterize the cluster. The assignment of samples to components is done by a maximum likelihood approach over the component posterior. That is, a sample x_i is assigned to component k^* such that

$$k^* = \underset{k}{\operatorname{argmax}}\, \tau_{ki}. \tag{2.26}$$

The component posterior τ_{ki} (Eq. (2.17)) captures the uncertainty of assignment of a given sample x_i [160]. If τ_{ik^*} is close to one, there is little ambiguity. One the other hand, a uniform component posterior means maximum uncertainty. This quantification of uncertainty in the cluster assignment can be visualized by computing the Shannon entropy [169] of the component posterior. Fig. 2.6 shows an example mixture density (2.6a) and the corresponding entropy of the component posterior (2.6b). It can be seen that the entropy is high in areas where clusters overlap and low towards the cluster centers. The highest entropy can be observed when all three clusters overlap.

One important consequence of the cluster assignment rule Eq. (2.26) is that it is invariant against deviations of the posterior which do not change k^*. In other words, for the cluster assignment of a sample to be correct it is sufficient that the true component obtains the highest posterior. This is one reason why the model to a certain degree is robust against the independence assumption not being met by the data.

2.3.1 Model Selection

One important aspect of clustering with mixture models is the choice of the number of components K. Classically this has been addressed by training mixtures with a range of components and then applying some model selection criterion to select the optimal number of components. Generally speaking, these criteria select models based on the principle of *maximum parsimony*, also known as *Occam's Razor* [6]. This principle stipulates that the simplest model which models the data sufficiently well should be used.

Criteria such as the *Bayesian information criterion* (BIC) [166] and the *Akaike information*

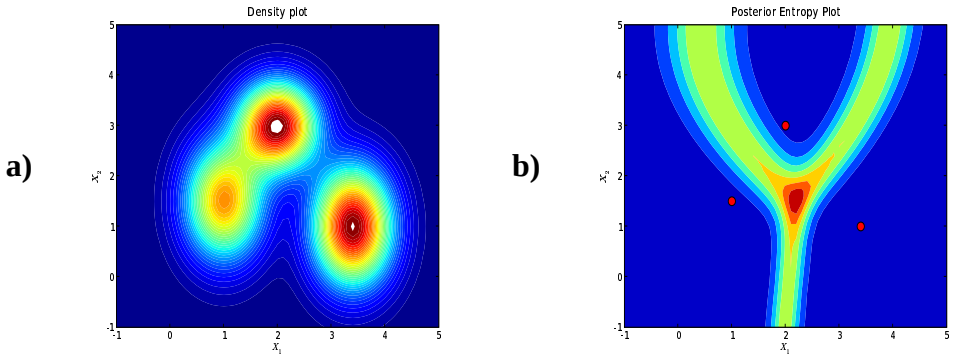


Figure 2.6: a) *Example mixture density plot.* ***b)*** *Entropy of the component posterior. The red dots mark the mode of the three components.*

criterion (AIC) [3] have been applied in this manner. BIC and AIC are penalized likelihood scores which contrast the likelihood of a model with its complexity. The BIC score for a maximum likelihood estimates $\hat{\Theta}$ is given by

$$BIC(\hat{\Theta}, D) = -2 \log L(\hat{\Theta}|D) + |\hat{\Theta}| \log(N), \tag{2.27}$$

where $L(\Theta|D)$ again is the likelihood function Eq. (2.14), $|\Theta|$ the number of free parameters in Θ and N the number of samples. Similarly, the AIC is defined as

$$AIC(\hat{\Theta}, D) = -2 \log L(\hat{\Theta}|D) + 2|\hat{\Theta}|. \tag{2.28}$$

It can be seen that the penalization for model complexity is stronger in the BIC as $\log(N) > 2$ for $N \geq 8$, which will be the case for most real world data sets. Therefore it can be said that the BIC is more conservative in the penalization of model complexity than AIC . In fact, in practice it is often observed that the BIC tends to underestimate the number of components, whereas the AIC tends to overestimate [61].

An alternative approach to model selection is taken by the *Normalized Entropy criterion* (NEC) [14, 61]. While the penalized likelihood scores are very general in concept, the NEC has been designed specifically for the choice of the number of components in a mixture. The NEC scores models by their ability to provide well-separated groupings of the data. The NEC arises from the decomposition of the $\log$–likelihood $\log L(\hat{\Theta}|D)$ in Eq. (2.14) into a $\log$-likelihood term and an entropy term. Since for the model selection problem we are only interested in the number of components K, we let $L_k = \log L(\hat{\Theta}_K|D)$ where $\hat{\Theta}_K$ is a mixture with K components.

Then it can be shown that

$$L_K = C_K - E_K, \tag{2.29}$$

where

$$C_K = \sum_{k=1}^{K} \sum_{i=1}^{N} \tau_{ki} \ \log[\pi_k \ P(x_i|\theta_k)] \tag{2.30}$$

and

$$E_K = -\sum_{k=1}^{K} \sum_{i=1}^{N} \tau_{ki} \ \log(\tau_{ki}). \tag{2.31}$$

C_K is a classification likelihood term and E_K measures the overlap between the different components. τ_{ki} again denotes the component posterior as introduced in equation Eq. (2.17).

Based on Eq. (2.29) the NEC is defined as

$$\text{NEC}(\Theta, K) = \frac{E_K}{L_K - L_1}. \tag{2.32}$$

A small value for $\text{NEC}(\Theta, K)$ is obtained for models which capture strong groupings in the data and the clusters are well separated. Since the NEC has been specifically designed for mixture model selection in clustering, in the following it is used as the model selection criterion of choice.

Alternative approaches for model selection include stability-based measures (e.g. [12, 103]), methods based on the classification likelihood [16], integrated complete likelihood [15, 16], Fisher information matrices [197] or methods based on random projections [50]. Recently Bayesian approaches for selecting the number of components have received some attention. These methods incorporate changes in K as an integral part of the parameter estimation, either as part of a MCMC procedure [152] or by modeling the prior over the mixture weights $P(\pi)$ as an stochastic process, in particular the Dirichlet process prior [48, 49, 126].

2.3.2 Clustering Evaluation

One possible setup for the validation of a clustering method is to compute the clustering in an unsupervised manner, and then contrast the cluster labels with the true, known class labels of a given data set. A clustering is given by a vector $c = (c_1, ..., c_N)$ and analogously the true labels $t = (t_1, ..., t_N)$, with $c_i, t_i \in 1, ..., K$. Since the mapping of clusters to classes is unknown, we cannot compare c and t directly. Instead, we consider all ordered pairs of samples $x_i, x_l, i > l$ and count whether the clustering correctly assigns the same or different labels. This leads to the number of true positives (TP) as

$$TP = \sum_{i=1}^{N} \sum_{l=i+1}^{N} \delta(t_i = t_l)\delta(c_i = c_l),$$

i.e. all pairs of samples where both the true labels and the cluster labels are the same for x_i and x_l. By the same reasoning we obtain the false positives (FP)

$$FP = \sum_{i=1}^{N} \sum_{l=i+1}^{N} \delta(t_i = t_l)\delta(c_i \neq c_l),$$

the true negatives (TN)

$$TN = \sum_{i=1}^{N} \sum_{l=i+1}^{N} \delta(t_i \neq t_l)\delta(c_i \neq c_l)$$

and the false negatives (FN)

$$FN = \sum_{i=1}^{N} \sum_{l=i+1}^{N} \delta(t_i \neq t_l)\delta(c_i = c_l).$$

Based on these quantities we can now compute the standard sensitivity, specificity and accuracy measures as

$$\text{Sensitivity} = \frac{TP}{TP + FN},$$

$$\text{Specificity} = \frac{TN}{TN + FP},$$

and

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

The sensitivity measures the fraction of comparisons where the clustering correctly assigned the same label, the specificity give the same for the correct assignment of unequal labels. The accuracy combines the two measures by given the total fraction of comparisons where the clustering matches the true labels.

2.3.3 Handling of Missing Data

Missing values are an issue which needs to be addressed for many real world data sets. If there are samples where a majority of the values are missing a drastic approach would be to excise them from the data set. However, even samples with many missing values may still retain some useful information and particularly in the case that there is little data, simply discarding samples might be wasteful. Another approach are so called *data imputation* techniques (e.g [88, 112, 144, 173]) where the missing values are replaced with values computed from the observed data and subsequently the data set is treated as being

complete. This however can introduce unwanted biases into the results, if an unsuitable imputation method is chosen [4]. In context of probabilistic models for clustering, this problem can be circumvented by explicitly accounting for missing data in the distributions used [161]. For discrete distributions this amounts to simply introducing a dedicated noise symbol m into the alphabet, i. e. $\Sigma = \Sigma \cup m$ For normally distributed, continuous data, handling of missing data is equivalent by re-defining the data range as $\mathbb{R} \cup m$. Then each atomic distribution in $P(x_i|\theta_k)$ is modified to assign some fixed probability $P(m)$ to the missing symbol, i.e.

$$P(x_{ij}|\theta_{kj}) = \begin{cases} P(x_{ij}|\theta_{kj}) & \text{if } x_{ij} \neq m \\ P(m) & \text{otherwise.} \end{cases}$$

The probability of the missing symbol m has to be specified *a priori* and does not change during the parameter estimation. This scheme has the effect that the missing values will yield the exact same probability under all components and therefore the contributions to the clustering will cancel out.

2.3.4 Dealing with Noisy Data Sets

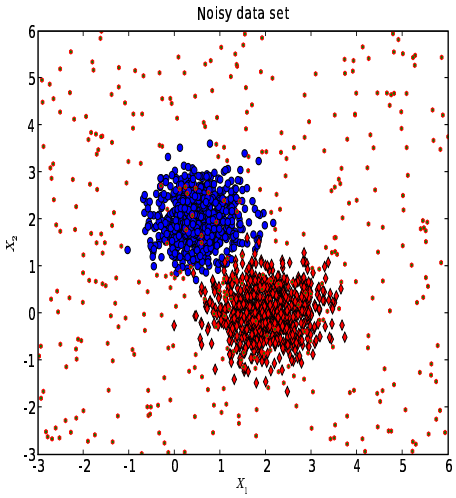


Figure 2.7: *Data set with two clusters and 20% noisy samples.*

Noise in the data is another common problem when dealing with real word data sets especially in bioinformatics. An example of a data set with two clusters (red and blue) and uniform noise (green dots) is shown in Fig. 2.7 for two–dimensional data (X_1, X_2). When attempting to fit a two component normal mixture to this data set, the noise will have a detrimental effect on the learned parameters. This can be seen in the density plot in the left part of Fig. 2.8. Here both components were basically merged due to the influence of the noise. One technique which is often useful in addressing this problem, is to explicitly

account for noise samples in form of dedicated, uniform noise components. This leads to the mixture

$$\sum_{k=1}^{K-1} \pi_k P(x_i|\theta_k) + \pi_K U(x_i),$$

where the K'th component is a naïve Bayes uniform distribution over X. The boundaries of each uniform distribution, as well as the fixed weight π_K are specified *a priori*.

In the example, the addition of a noise component over the range of values observed in the data leads to the model on the right in Fig. 2.8. Here both clusters have been captured correctly. It should be noted that for this example the true distribution of the noise was indeed uniform and therefore the noise component provided a perfect fit. However uniform noise component retain their usefulness also for the situation where the true distribution of the noise is unknown [8, 84].

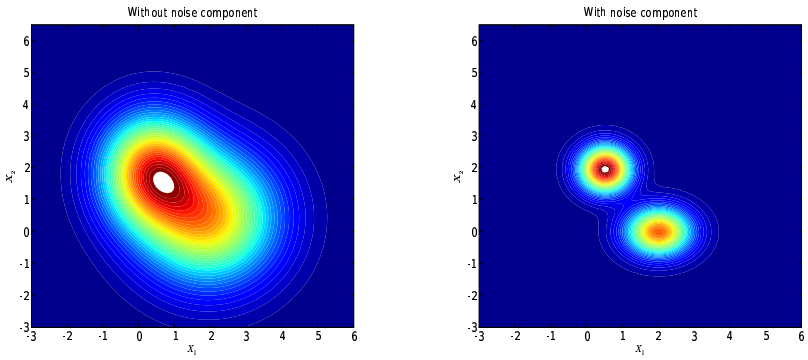


Figure 2.8: *Left: Mixture estimates on the noisy data set for a two component mixture. Right: Mixture estimates on the same data set with the addition of uniform noise component.*

Another technique to reduce the impact of noise in the data is *deterministic annealing* [157]. In *deterministic annealing* the assignment of samples to components by the component posterior Eq. (2.17) during the EM procedure is shifted toward the uniform distribution. The shift is reduced gradually over successive iterations until the algorithm continues normally. This has the effect of potentially avoiding poor local maxima in the likelihood. The procedure can be seen as special case of a *simulated annealing* setup [91].

2.4 Bayesian Mixture Models

Bayesian statistics deal with the integration of prior knowledge into the process of inference over a data set [63]. This prior knowledge decreases uncertainty about the model parameters and causes a regularization of the parameter estimates. The inclusion of prior expert knowledge for a specific application can also help to achieve more meaningful results. The former is realized in the Bayesian framework by new parameter estimators which take prior knowledge into account. An example for the latter will be described in chapter 6.

This prior knowledge comes in form of prior distributions $P(\Theta|M)$ over the model parameters where M is the model class, i.e. the number of components of the mixture. The joint likelihood of D and Θ is given by

$$P(D, \Theta|M) = P(D|\Theta, M)P(\Theta|M) \tag{2.33}$$

with the mixture likelihood again given by

$$P(D|\Theta, M) = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k \prod_{j=1}^{p} P(x_{ij}|\theta_{kj}) \tag{2.34}$$

and the parameter prior

$$P(\Theta|M) = P(\pi) \prod_{k=1}^{K} \prod_{j=1}^{p} P(\theta_{kj}). \tag{2.35}$$

This means that due to the independence between parameters, the prior for the whole mixture decomposes into the product of prior terms for each individual parameter in Θ. Details for the individual priors will be given in the following section.

The parameter estimation task is then to find the Θ which maximizes the joint likelihood. These *maximum a posteriori* (MAP) parameters $\hat{\Theta}$ are given by

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}}\, P(D, \Theta|M).$$

That is, Θ is estimated with the value in the mode of the joint likelihood of D and Θ. A distinct advantage of MAP estimation is that it can be straightforwardly integrated into the EM framework and that all convergence results mentioned in section 2.2 still apply. The objective function of the MAP EM is obtained by substituting Eq. (2.33) into Eq. (2.18) (see section 2.4.2).

It must be stressed that the approach taken here is not fully Bayesian in that we do not evaluate the marginal probability of the data. Rather the approach taken is equivalent to a penalized maximum likelihood estimation.

One problem with the full Bayesian approach is that the marginal cannot be computed analytically due to the incomplete data setting, although approximations exist [31]. Alternative approaches for the evaluation of the marginal likelihood are Markov chain Monte Carlo (MCMC) sampling techniques [155] which sample directly from the posterior. This class of methods includes techniques such as Gibbs and importance sampling [30, 71, 135, 151]. While properly applied these methods are fairly accurate, the lack of efficiency often limits their practical usability for real world data sets. Also, variational methods [131, 190] can be applied. These types of approaches define tractable bounds on the likelihood function and operate by maximizing these bounds rather than the likelihood directly.

One important advantage of the MAP approach is that it allows the calculation of the posterior distribution of a model M and the corresponding MAP parameters $P(M, \hat{\Theta}|D)$,

i.e. the distribution over different models $M, \hat{\Theta}$ given data set D. The model posterior is obtained by *Bayes rule* as

$$P(\hat{\Theta}, M|D) = \frac{P(D|\hat{\Theta}, M)P(\hat{\Theta}, M)}{P(D)} = \frac{P(D|\hat{\Theta}, M)P(\hat{\Theta}|M)P(M)}{P(D)}, \tag{2.36}$$

where the model prior $P(M)$ is a penalizing factor for model complexity (see chapter 3 for more details). Eq. (2.36) can be simplified to

$$P(\hat{\Theta}, M|D) \propto P(D|\hat{\Theta}, M)P(\hat{\Theta}|M)P(M). \tag{2.37}$$

This is an important simplification as it allows us to score different M without having to evaluate the term $P(D)$ which would require integration over all possible models. The parameter estimation task can then be stated to find the model M which maximizes the model posterior for a given data set D.

2.4.1 Conjugate Priors

The standard choices for parameter priors $P(\theta_{kj})$ are the respective conjugate priors for the distributions in Θ. A prior over θ_{kj} defines a distribution over the parameter space of θ_{kj}. The prior distributions are parameterized by hyperparameters. A conjugate prior has the property that the posterior $P(\theta_{kj}|D)$ has the same distribution as the prior, but with hyperparameters updated according to the data.

The conjugate prior for discrete distributions is the Dirichlet distribution. The Dirichlet distribution defines a density over the space of stochastic vectors ϕ with dimension M. The distribution is parameterized by a vector of hyper-parameters $\alpha = (\alpha_1, ..., \alpha_M), \alpha_s > 0$. The density function is given by

$$P(\phi|\alpha) = \frac{\Gamma(\sum_{s=1}^{M} \alpha_s)}{\sum_{s=1}^{M} \Gamma(\alpha_s)} \prod_{s=1}^{M} \phi_s^{\alpha_s - 1}, \tag{2.38}$$

where Γ is the Gamma function [1]. Fig. 2.9a) shows an example Dirichlet density for the three-dimensional simplex with parameters $\alpha = (1.5, 1.5, 1.5)$. The mode of the density is achieved for uniform discrete distributions. For an example of modeling prior knowledge using the Dirichlet distribution, refer to chapter 6.2.

The conjugate prior for the univariate normal distribution $N(\mu, \sigma^2)$ is given by the Normal-Inverse-Gamma prior. This prior takes the form

$$P(\mu, \sigma^2|\mu_p, \kappa_p, \varsigma_p, \nu_p^2) = P(\mu|\mu_p, \sigma^2/\kappa_p)P(\sigma^2|\varsigma_p, \nu_p^2), \tag{2.39}$$

where the first term on the right side is the normal density (Eq. 2.4) for mean μ_p and

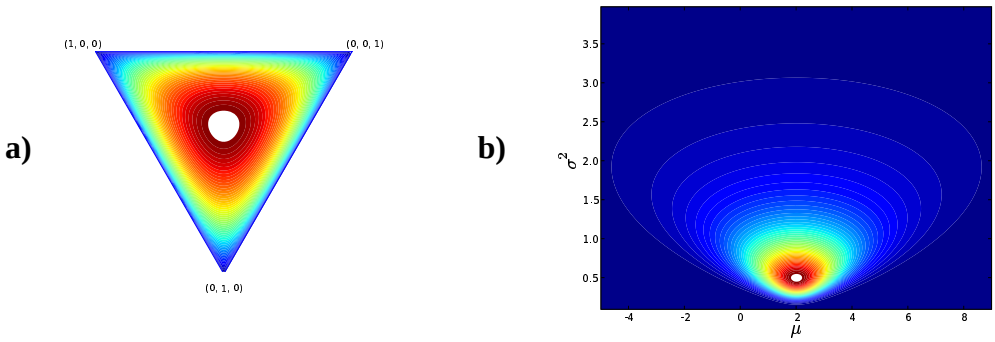


Figure 2.9: *a) Density plot for the Dirichlet distribution with* $M = 3$ *and* $\alpha = (1.5, 1.5, 1.5)$ *b) Normal-Inverse-Gamma density plot for* μ *and* σ^2 *with parameters* $\mu_p = 2, \kappa_p = 0.5, \varsigma_p = 2, \nu_p = 2$

variance σ^2/κ_p and the second term is the Inverse-Gamma distribution given by

$$P(\sigma^2|\varsigma_p, \nu_p^2) = \frac{\nu_p^{2\varsigma_p}}{\Gamma(\varsigma_p)}(\sigma^2)^{-\varsigma_p+1}\exp\left(-\frac{\nu}{\sigma^2}\right). \tag{2.40}$$

An example for the prior density over μ and σ^2 defined by this prior is shown in Fig. 2.9b). It can be seen that the density is zero for very small values of σ^2. This helps alleviate the problem of vanishing variances during the EM parameter estimation (see section 2.2.4).

2.4.2 Parameter Estimators

The MAP estimators differ from the ML estimators given in section 2.2.3 by the contribution of the parameter priors. Substituting the joint likelihood Eq. (2.33) into the EM Q function Eq. (2.18) leads to partial derivatives of the Lagrangian for the MAP estimators for π for the conjugate Dirichlet prior

$$\frac{\delta Q}{\delta \pi_k}\left[\sum_{k=1}^{K}\sum_{i=1}^{N}\tau_{ki}\log\pi_k + \log\frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\sum_{k=1}^{K}\Gamma(\alpha_k)}\sum_{k=1}^{K}(\alpha_k - 1)\log\pi_k + \lambda(\sum_{k=1}^{K}\pi_k - 1)\right] =$$
$$\frac{1}{\pi_k}(\sum_{i=1}^{N}\tau_{ki} + (\alpha_k - 1)) + \lambda = 0,$$
$$\frac{\delta Q}{\delta \lambda}\left[\sum_{k=1}^{K}\pi_k - 1\right] = 0.$$

Similarly to the derivation of the MLE case (section 2.2.3), solving for λ yields

$$\lambda = -(N + |\alpha| - K)$$

and consequently the MAP estimator for π under the Dirichlet prior as

$$\hat{\pi}_k = \frac{\sum_{i=1}^{N} \tau_{ki} + \alpha_k - 1}{N + |\alpha| - K} \quad (k = 1, ..., K). \tag{2.41}$$

Thus, the estimator for a discrete distribution $\theta_{kj} = \phi_{kj}$ with $\phi = (\phi_{kj1}, ..., \phi_{kjM})$ over some M symbol alphabet Σ is given by

$$\hat{\phi}_{kjs} = \frac{\sum_{\substack{i=1, \\ x_{ij}=\Sigma_s}}^{N} \tau_{ki} + \alpha_k - 1}{\sum_{i=1}^{N} \tau_{ki} + |\alpha| - M}, \quad (s = 1, ..., M). \tag{2.42}$$

Mathematically, the Dirichlet MAP estimators are equivalent to adding $|\alpha|$ observations to the data set, where each symbol is observed with frequency α_r. This gives a direct intuition on how the prior knowledge expressed by the prior influences the parameter estimation in the MAP setting.

For a normal distribution $\theta_{kj} = (\mu_{kj}, \sigma_{kj}^2)$ under the Normal-Inverse-Gamma prior, the MAP estimates for the mean μ_{kj} parameter are given by

$$\hat{\mu}_{kj} = \frac{\sum_{i=1}^{N} \tau_{ki} x_{ij} + \kappa_p \mu_p}{\sum_{i=1}^{N} \tau_{ki} + \kappa_p}, \tag{2.43}$$

that is the prior contribution adds κ_p observations with value μ_p.

For the variance σ_{kj}^2 MAP estimates are obtained as

$$\hat{\sigma}_{kj}^2 = \frac{\sum_{i=1}^{N} \tau_{ki}(x_i - \bar{\mu}_{kj})^2 + \varsigma_p^2 + \frac{\kappa_p n_k}{\kappa_p + n_k}(\bar{\mu}_{kj} - \mu_p)^2}{\sum_{i=1}^{N} \tau_{ki} + \nu_p + 3}. \tag{2.44}$$

with $n_k = \sum_{i=1}^{N} \tau_{ki}$ and $\bar{\mu}_{kj}$ given by the ML estimator for μ_{kj} in Eq. (2.23). See [53] for details on the derivation.

2.5 Partially-supervised learning

Clustering is performed in an *unsupervised* setup. This means that the assignment of samples to clusters is unknown *a priori*. A variant of this problem is the case where for a subset of samples there is information about the cluster memberships. A biological example for such a situation would be a set of protein sequences some of which have functional annotations. This is referred to as a *partially-supervised* (or also *semi-supervised*) learning [29, 160] problem. This information usually takes the form of positive (must-link) or negative (must-not-link) constraints for pairs of samples.

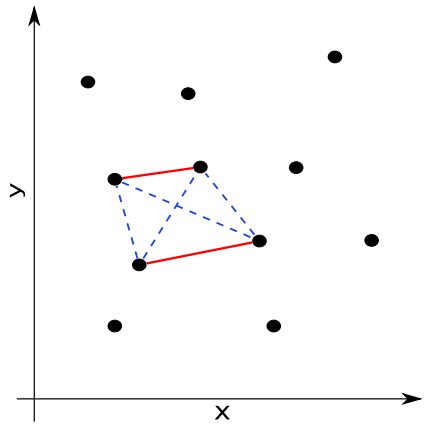


Figure 2.10: *Example of constraints arising from labeled samples. The data is two–dimensional with features x and y. Red edges between points indicate positive* must-link *constraints, blue edges negative* must-not-link *constraints.*

Fig. 2.10 shows an example for the constraints implicit in the labeling of data samples. Red edges between points represent *must-link* constraints, where each red edge stands for a different label, blue dashed edges *must-not-link* constraints. Each positive constraint implies negative constraints to all data points constrained to different clusters.

A simple way to implement positive constraints in the mixture case is to fix the assignment of samples with positive constraints to the same component in the component posterior (Eq. (2.17)). That is, for a labeled sample x_i with label l this means $\tau_{ki} = 1$ for $k = l$ and 0 for all other k. This binds the contribution of the sample to parameter estimation to a specific component. This setup can also be thought of as a point in the continuum between complete data and incomplete data learning tasks. For the former, the assignment of samples to components is known (i.e. the posterior takes the form given above). For the latter, the assignment of samples to components is unknown and the EM algorithm needs to be used to arrive at estimates for Θ.

This simple modification described above gives rise to the partially-supervised EM algorithm for estimation of Θ with hard positive constraints. More complex variants of the partially-supervised setup have been explored in the literature, including negative constraints and soft constraints [38, 102, 167].

Chapter 3

Context-specific Independence Mixture Models

The concept of *context-specific* independence (CSI) arose in the setting of Bayesian networks. It should be kept in mind that finite mixture models can be seen as a special case of a Bayesian network with constrained graph topology (see section 2.1.2).

Bayesian networks define a structured decomposition of the joint distribution of a collection of RVs. Such a decomposition is required, since the full-conditional distribution (where each RV is dependent on all the others) has a number of parameters exponential in the number of RVs and in general cannot be estimated from finite samples. It is therefore crucial to limit the model complexity (loosely the number of free parameters) by making assumptions about structural regularities in the joint distribution. In case of Bayesian networks one such assumption would be the Markov property, i.e. that each RV is only dependent on its parents in the network structure. In a finite data setting, attempting to learn too complex a model will lead to overfitted parameters and spurious results. A good model can then be characterized as being simple enough to allow robust inference while at the same time capturing the most relevant trends in the data.

The central idea of the CSI formalism is to increase robustness by making use of regularities in the parameters of a model to reduce and adapt model complexity to the degree of variability observed in the data.

3.1 Prior Work

This notion of reducing model complexity and enhancing inference by capturing additional structure in the model parameters has received considerable attention in the Bayesian networks community. In addition to CSI [22, 32, 57] this includes approaches such as *similarity networks* [82], *multinets* [62], asymmetric representations for decision making [172], decision trees [21, 70] and the application of probabilistic Horn rules [145].

The first application of CSI in the context of mixture models was using Bayesian CSI mixture models for clustering gene expression data [10]. The main difference between [10]

and the model we are going to introduce in this chapter lies in the formulation of CSI used. [10] used the so called *default tables* [57], which allows for far less flexibility in the CSI structure than our formulation (see section 3.3 for details). This new CSI formulation has profound impacts on the usefulness of the model for practical data analysis.

We previously applied our CSI mixtures for the analysis of attention deficit hyperactivity disorder patient data [64]. However, this work did not take advantage of the Bayesian framework and the structure learning was done by a fairly simplistic AIC-based clustering of model parameters following the EM estimation.

3.2 Context-specific Independence (CSI)

Formally, *statistical independence* for two RVs X_j and C is defined as

$$P(X_j|C) = P(X_j). \tag{3.1}$$

In the mixture framework, each of the $X_1, ..., X_p$ is dependent on the component RV C which takes values k in the set of component indices $(1, ..., K)$. The dependency on C is then expressed by the component-specific parameters θ_{kj}. This leads to the mixture distribution familiar from Eq. (2.1) and (2.2)

$$P(x_i|\Theta) = \sum_{k=1}^{K} \pi_k P(x_{i1}|\theta_{k1}) P(x_{i2}|\theta_{k2}) ... P(x_{ip}|\theta_{kp}), \tag{3.2}$$

where the sum goes over the possible values of C, i.e. $(1, ..., K)$. Intuitively, this dependence on the component variable C follows the assumption that a feature carries regularities which help to characterize and discriminate the components. Conversely, a feature which does not contain such information is *uninformative* for the purpose of characterizing the components and therefore *independent* of C. This is not an uncommon situation especially for exploratory data analysis where the relevance of each feature for the clustering is not known *a priori*. If we assume without loss of generality that feature X_1 is independent of C, i.e. $P(x_{i1}|\theta_{k1}) = P(x_{i1})$ for all $k \in (1, ..., K)$ we have

$$P(x_i|\Theta) = P(x_{i1}) \sum_{k=1}^{K} \pi_k P(x_{i2}|\theta_{k2}) ... P(x_{ip}|\theta_{kp}). \tag{3.3}$$

The kind of model in Eq. (3.3), where one or several features have been set to be independent of the component variable, has been referred to as *selective naive Bayes* mixtures [10]. Essentially, feature X_1 equally contributes to the likelihood of each component and therefore has no impact on the component membership posterior (Eq. (2.17)) or the cluster assignments.

In between the two extremes of full in– or dependence on C is the case where some feature,

say X_1, is informative for characterizing and discriminating subsets of components. For instance a feature may discriminate two rather broad categories of clusters whereas other features contain information for division into further subcategories. In such a case, the natural parameterization is to identify a separate set of parameters with each subset of components, i.e. the *context* the feature discriminates.

This notion of *context-specific* independence is formalized as an extension to the conventional mixture framework in the next section.

3.3 CSI for Mixture Models

In case of conventional mixture models the *conditional independence* assumption between components means that for each value of the component variable C, a separate set of parameters θ_{kj} needs to be specified. In the CSI case, several components may share parameters in a feature, depending on different contexts, i.e. subsets of C.

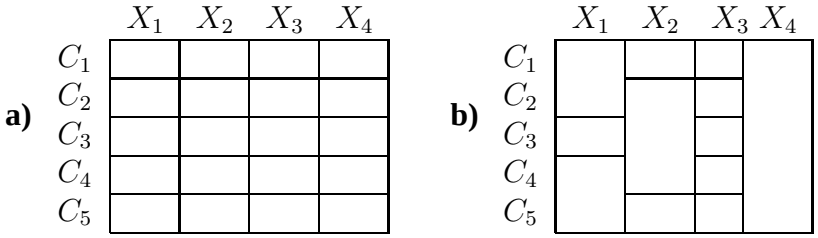


Figure 3.1: ***a)*** *Model structure for a conventional mixture with 5 components and four RVs. Each cell of the matrix represents a distribution in the mixture and every RV has an unique distribution in each component.* ***b)*** *CSI model structure. Multiple components may share the same distribution for a RV as indicated by the matrix cells spanning multiple rows. In example C_2, C_3 and C_4 share the same distribution for X_2.*

From the structure matrix for a conventional mixture shown in Fig. 3.1a) this leads to a CSI structure shown in Fig. 3.1b). Again, each cell of the matrix represents an uniquely parameterized distribution but several components may share parameters for certain features. This means that for example C_1 and C_2 are represented by the same distribution for X_1 and all components share the same distribution for X_4. It should be noted that the visual structure matrix representation used here is a simplification since it can only represent groups which are contiguous in a column. However, for the sake of the example this is sufficient and if full generality is required, the different groups in the matrix can be color coded (see section 3.6.1).

In addition to the reduction in model complexity, the structure matrix also helps to facilitate a cluster analysis by giving an explicit, high-level overview of the regularities in the data which characterize the different components (i.e. clusters). For instance, in the example one can see that for feature X_1 components (C_1, C_2) and (C_4, C_5) share characteristics and are represented by one set of parameters. On the other hand component C_3 does not share its parameterization for X_1. Moreover, if components share the same group in the CSI

structure for all positions, they can be merged thus reducing the number of components in the model. Therefore learning of a CSI structure can amount to an automatic reduction of the number of components as an integral part of model training.

We will further discuss the practical implications of the CSI structure for cluster analysis in section 3.6.1.

The *default table* representation of CSI used in [10] assumes that for each column there is only a single group in the CSI matrix with size larger than one. In the example, this means that the structures of X_2, X_3 and X_4 can be represented as *default tables*, whereas X_1 cannot. As the number of components increases, so does the restrictions on structure space imposed by the *default table* representation.

The more general formulation of CSI structures used in this work allows for the capture of more and richer regularities in the data, which enhances the usefulness of CSI for facilitating data analysis. This will become apparent when applying the model to biological data sets (esp. chapter 7.

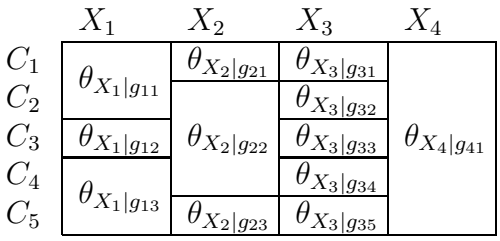


Figure 3.2: *CSI parameter matrix for the structure shown in Fig. 3.1a).*

Formally, we define the CSI mixture model as follows: For the set of component indexes $\mathcal{C} = \{1, .., K\}$ and variables $X_1, ..., X_p$, let $G = \{g_j\}_{(j=1,...,p)}$ be the CSI structure of the model M. Then $g_j = (g_{j1}, ...g_{jZ_j})$ where Z_j is the number of subgroups for X_j and each $g_{jr}, r = 1, ..., Z_j$ is a subset of component indexes from $\mathcal{C}$. That is, each g_j is a partition of $\mathcal{C}$ into distinct subsets where each g_{jr} represents a subgroup of components which share the same distribution for X_j. The CSI mixture distribution is then obtained by replacing $P(x_{ij}|\theta_{kj})$ with $P(x_{ij}|\theta_{X_j|g_j(k)})$ in (2.1) where $g_j(k) = g_{jr}$ such that $k \in g_{jr}$.

This yields the mixture distribution as

$$P(x_i|\Theta) = \sum_{k=1}^{K} \pi_k \prod_{j=1}^{p} P(x_{ij}|\theta_{X_j|g_j(k)}), \tag{3.4}$$

where accordingly $\Theta = (\pi, \theta_{X_1|g_1}, ..., \theta_{X_1|g_{Z_1}}, ..., \theta_{X_p|g_1}, ..., \theta_{X_p|g_{Z_p}})$ is the model parameterization.

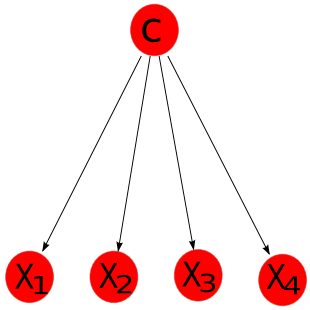
3.3.1 CSI from Different Perspectives

In this section we revisit the two different perspectives on mixture models described in section 2.1.2 (Bayesian networks and HMMs and examine how the adoption of the CSI formalism is reflected in these models.

CSI in Bayesian Networks

While the finer-grained statements of CSI cannot be represented in the canonical Bayesian network graph [22], they become apparent by regularities in the CPT tables. Fig. 3.3 shows again the example network graph and the CSI CPT for feature X_1.

a)



b)

C	X_2
1	$\theta_{X_1\|g_{11}}$
2	$\theta_{X_1\|g_{11}}$
3	$\theta_{X_1\|g_{12}}$
4	$\theta_{X_1\|g_{13}}$
5	$\theta_{X_1\|g_{13}}$

Figure 3.3: *a) Bayesian network graph for a mixture distribution with four features. $X_1 - X_4$ are conditionally independent given the component indicator C. b) CPT for feature X_1. The table contains parameters according to the CSI structure shown in Fig. 3.1a).*

It can be seen that for contexts $c \in ((1,2),(3),(4,5))$ X_1 has a specific distribution . Also, the relation to the corresponding CSI mixture parameter matrix Fig. 3.2b) is obvious.

CSI in Hidden Markov Models

From the perspective of an HMM, the CSI structure amounts to a tying of state emission and transition distribution parameters. The ideas of reducing model complexity in HMMs by representing several states with the same distribution [159, 183], and of adapting the model complexity by topology learning [178], have received some attention in the literature.

The HMM topology for the CSI structure matrix in Fig. 3.1b) is shown in Fig. 3.4. Here boxes around states imply a tying of the corresponding emission and transition parameters.

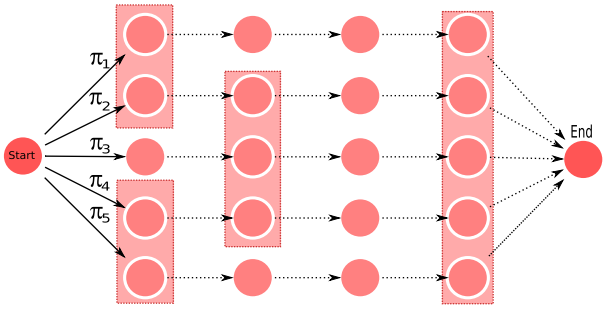


Figure 3.4: *HMM topology with state parameter tying equivalent to the five component CSI mixture from Fig. 3.1.*

3.4 Bayesian CSI Mixtures

To extend the model posterior (Eq. (2.37)) to the CSI case, the model prior $P(M)$ is set as a prior term over the CSI structure. Again we have the model posterior Eq. (2.37)

$$P(\hat{\Theta}, M|D) \propto P(D|\hat{\Theta}, M)P(\hat{\Theta}|M)P(M)$$

with $P(D|\hat{\Theta}, M)$ given by adaptation of Eq. (2.34) for CSI mixtures. The prior over the CSI structure $P(M)$ serves as a regularizer for the structure learning similar to the parameter priors when computing parameter estimates. Any prior knowledge about the CSI structure of a given model can be encoded in this prior. In general though, there exists no such prior knowledge and the prior only captures a general preference for a less complex model. For such a prior $P(M)$ we adopted the factored form

$$P(M) \propto P(K)P(G), \tag{3.5}$$

where the $P(K)$ is the prior over the number of components and $P(G)$ is the model structure prior defined as

$$P(K) = \gamma^K, \qquad P(G) = \prod_{j=1}^{p} \omega^{Z_j} \tag{3.6}$$

where $\gamma, \omega < 1$ are hyperparameters, p is the number of dimensions, K is the number of components and Z_j is the number of groups in the CSI structure of feature X_j. That is, both prior terms will decrease for larger K and Z_j (due to $\gamma, \omega < 1$) and thereby penalize complex models. Therefore by means of the prior a bias towards smaller models and simpler structures is introduced into the model posterior. The values of γ and ω need to be chosen a priori and can be considered to adapt the strength of the preference for a less complex structure. It should be noted that, of the two, ω is the more important hyperparameter as it drives the feature-wise structure learning, whereas γ only contributes to the posterior for structures where entire components are merged together in the structure. In

section 4.3.1, we give a simple data-driven heuristic to choose the hyper-parameters for a given data set.

3.5 Structural EM Algorithm

In the previous sections, we introduced the CSI formalism, and motivated it by describing both its desirable properties for robustness in parameter estimation and facilitation of cluster analysis. In order to harness these advantages in practice, a reliable way to learn such a CSI structure from data is required. Our structure learning method of choice is the structural EM algorithm (sEM) [55, 56]. The sEM framework is an extension of the classic parametric EM (section 2.2) in which the unknown CSI structure is inferred based on the expected sufficient statistics of the data given the structure.

3.5.1 General Formulation

Before describing the structure learning for CSI mixtures in detail, we give the extensions to the parametric EM which give rise to sEM generically. This entails adapting the EM Q function (Eq. (2.6)) for the CSI case. In addition to the parameters Θ, the structure learning requires assessment of a structure G as part of a model M. We consider cases where the objective scoring metric takes the form

$$S(\Theta, M) = \log P(D|\Theta, M) - \text{Pen}(\Theta, M), \tag{3.7}$$

where $\text{Pen}(\Theta, M)$ is a penalty function based on the current parameters Θ and the model M. This formulation includes classic model selection criteria such as BIC(Eq. (2.27)) or AIC (Eq. (2.28)) as well as the Bayesian mixture model posterior Eq. (2.37).

In analogy to the parametric EM, the sEM objective Q-function is the expectation of the scoring metric

$$Q(\Theta, M; \Theta^{t-1}, M^{t-1}) = E[\log P(D, H|\Theta, M) - \text{Pen}(\Theta, M)|D, \Theta^{t-1}, M^{t-1}]. \tag{3.8}$$

The expectation can be computed by taking the integral over all possible values of the hidden data $h \in H$ and $P(h|D, \Theta^{i-1}, M^{i-1})$ is the distribution of the hidden data.

$$E[\log P(D, H|\Theta, M) - \text{Pen}(\Theta, M)|D, \Theta^{t-1}, M^{t-1}] =$$
$$\int_{h \in H} \log(P(D, h|\Theta, M) - \text{Pen}(\Theta, M))\, P(h|D, \Theta^{t-1}, M^{t-1})\, dh. \tag{3.9}$$

In analogy to the parametric EM procedure [128], it can be shown [55] that, by choosing

model and parameters which maximize the Q function

$$(\Theta^t, M^t) = \underset{\Theta, M}{\operatorname{argmax}}\, Q(\Theta, M; \Theta^{t-1}, M^{t-1}),$$

the increase in the Q function in each step, i.e.

$$Q(\Theta^t, M^t; \Theta^{t-1}, M^{t-1}) > Q(\Theta, M; \Theta^{t-1}, M^{t-1})$$

also guarantees that the score itself increases

$$S(\Theta^t, M^t) > S(\Theta^{t-1}, M^{t-1}).$$

This holds until convergence is reached [55] and there is no more change in the score, i.e. $S(\Theta^{t+1}, M^{t+1}) = S(\Theta^t, M^t)$.

3.5.2 Structural EM for Bayesian CSI Mixture Models

The task of learning a CSI model from data consists of assigning values to the group structure variables g_j and estimating parameters Θ for the induced distributions.

We adopt the Bayesian approach described in section 3.4 for the scoring function, that is different models are scored by the model posterior distribution (Eq. (2.37))

$$P(\hat{\Theta}, M|D) \propto P(D|\hat{\Theta}, M)P(\hat{\Theta}, M) = P(D|\hat{\Theta}, M)P(\hat{\Theta}|M)P(M)$$

where again $P(D|\hat{\Theta}, M)$ is the likelihood based on the data D (Eq. (2.34)), $P(\hat{\Theta}|M)$ is the parameter prior (Eq. (2.35)), $P(M)$ is the prior over the model structure (Eq. (3.5)) and the $\hat{\Theta}$ are the MAP parameter estimates (Eq. (2.4)).

From Eq. (3.7) the sEM Q function for the mixture case is given by

$$Q(\Theta^t, M^t; \Theta^{t-1}, M^{t-1}) = E[\log P(D|\Theta, M) - \log P(\Theta, M)|\Theta^{t-1}, M^{t-1}]. \qquad (3.10)$$

In the next section the parameter estimators for the sEM algorithm will be derived by taking derivatives of Eq. (3.10) with respect to the model parameters.

3.5.3 Structure Parameter Estimators

Using Eq. (2.18) and Eq. (3.4) we can write the CSI Q function Eq. (3.10) as

$$Q(\Theta, M; \Theta^{i-1}, M^{i-1}) =$$
$$\sum_{k=1}^{K}\sum_{i=1}^{N} \tau_{ki} \left(\log \pi_k + \sum_{j=1}^{p} \log P(x_{ij}|\theta_{X_j|g_j(k)}) \right) + \log P(\Theta|M) + \log P(M). \qquad (3.11)$$

The estimators for π remain unchanged from Eq. (2.41). When taking derivatives with respect to some $\theta_{X_j|g_{jr}}$, the difference to the conventional MAP case is that in the sum over K there are contributions from all $k \in g_{jr}$. After substituting the prior $\log P(\Theta|M)$ (Eq. (2.35)), taking the derivative with respect to a given $\theta_{X_j|g_{jr}}$ and setting to zero we have

$$\frac{\delta Q}{\delta \theta_{X_j|g_{jr}}} \left[\sum_{k=1}^{K} \sum_{i=1}^{N} \tau_{ki} \left(\log \pi_k + \sum_{j=1}^{p} \log P(x_{ij}|\theta_{X_j|g_j(k)}) \right) + \log P(\pi) + \sum_{k=1}^{K} \sum_{j=1}^{p} \log P(\theta_{X_j|g_j(k)}) + \log P(M) \right] = 0.$$

This can be simplified by dropping terms independent of feature j to

$$\frac{\delta Q}{\delta \theta_{X_j|g_{jr}}} \left[\sum_{k=1}^{K} \sum_{i=1}^{N} \tau_{ki} \log P(x_{ij}|\theta_{X_j|g_j(k)}) + \sum_{k=1}^{K} \log P(\theta_{X_j|g_j(k)}) \right] = 0. \tag{3.12}$$

The general formulation in Eq. (3.12) can be adapted for specific atomic distributions $\theta_{X_j|g_{jr}}$ by substituting the density function $P(x_{ij}|\theta_{X_j|g_j(k)})$ and the conjugate prior density $P(\theta_{X_j|g_j(k)})$. For instance, let $\theta_{X_j|g_{jr}}$ be a discrete distribution, i.e. $\theta_{X_j|g_{jr}} = \phi_{g_{jr}} = (\phi_{g_{jr}1}, ..., \phi_{g_{jr}M})$ with $\phi_{g_{jr}}$ being stochastic. For ease of reading, in the following we drop the index g_{jr} from $\phi_{g_{jr}}$, i.e. $\phi_{g_{jr}} = \phi$. Then the derivative of the Lagrangian with respect to the elements of ϕ and λ are

$$\frac{\delta Q}{\delta \phi_s} \left[\sum_{k=1}^{K} \sum_{i=1}^{N} \tau_{ki} \log \phi_{x_{ij}} + \log \sum_{k=1}^{K} \frac{\Gamma(\sum_{s=1}^{M} \alpha_s)}{\sum_{s=1}^{M} \Gamma(\alpha_s)} \prod_{s=1}^{M} \phi_s^{\alpha_s - 1} + \lambda (\sum_{s=1}^{M} \phi_s - 1) \right] =$$

$$\frac{1}{\phi_s} \left(\sum_{k \in g_j(r)} \sum_{\substack{i=1, \\ x_{ij} = \Sigma_s}}^{N} \tau_{ki} + (\alpha_s - 1) \right) + \lambda = 0, \tag{3.13}$$

$$\frac{\delta Q}{\delta \lambda} \left[\sum_{s=1}^{M} \phi_s - 1 \right] = 0. \tag{3.14}$$

Analogously to the derivation of the ML (section (2.2.3)) and MAP (section (2.4.2)) estimators, the CSI mixture MAP estimators are obtained by solving the Lagrangian for ϕ and λ. In this case we obtain $\lambda = -(\sum_{k \in g_j(r)} \sum_{i=1}^{N} \tau_{ki} + Z_j(\alpha_s - M))$ where Z_j again is the size of the current group structure g_{jr}. Note that, due to the summation over M in Eq. (3.14), the conditioning on a specific symbol Σ_s in Eq. (3.13) drops out of the sum over N.

For discrete distributions ϕ, the parameter estimators for ϕ_s and structure g_{jr} is then given

by

$$\hat{\phi}_s = \frac{\left(\sum_{k \in g_j(r)} \sum_{\substack{i=1, \\ x_{ij}=\Sigma_s}}^{N} \tau_{ki}\right) + Z_j(\alpha_s - 1)}{\sum_{k \in g_j(r)} \sum_{i=1}^{N} \tau_{ki} + Z_j(|\alpha| - M)} \quad (s = 1, ..., M). \tag{3.15}$$

When contrasting Eq. (3.15) with the results for the MAP estimators in Eq. (2.42), it can be seen that the estimator for a given CSI structure is obtained by pooling, i.e. adding up, the component posterior and prior contributions of all components in the group. This makes intuitive sense when thinking of the model posterior τ_{ki} as the relative contribution of sample x_i to the parameters of component k. In order to obtain the parameters for the case where components share the same distribution for a feature X_j, we pool the contributions of the components. This pooling of the posterior gives rise to the *expected sufficient statistics* of the data given the structure in the estimators. One important consequence of this result is that once we have computed the model posterior for each component k separately, we can get estimates for all possible groupings in the CSI structure in an efficient manner by pooling the posterior over subsets of components.

The estimators for a Gaussian $\theta_{X_j|g_{jr}} = (\mu_{g_{jr}}, \sigma^2_{g_{jr}}) = (\mu, \sigma^2)$ (again we drop the indices) are obtained by a straightforward extension of the derivation given in [53]. First, we substitute the Gaussian (Eq. (2.4)) and and Normal-Inverse-Gamma prior Eq. (2.39) densities into Eq. (3.12) to obtain

$$\begin{aligned}
&\sum_{k=1}^{K} \sum_{i=1}^{N} \tau_{ki} \left[\log \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x_{ij} - \mu)^2}{2\sigma^2} \right) \right] + \\
&\sum_{k=1}^{K} \log \left[\frac{1}{\sqrt{(\sigma^2/\kappa_p)2\pi}} \exp\left(-\frac{\kappa_p}{2\sigma^2}(\mu - \mu_p)^2 \right) \right] \left[\frac{\nu_p^{2\varsigma_p}}{\Gamma(\varsigma_p)} (\sigma^2)^{-\varsigma_p+1} \exp\left(-\frac{\nu}{\sigma^2} \right) \right] = 0.
\end{aligned} \tag{3.16}$$

Taking the derivative with respect to μ of Eq. (3.16) yields

$$\begin{aligned}
\frac{\delta Q}{\delta \mu} &= \sum_{k \in g_j(r)} \sum_{i=1}^{N} \tau_{ki} \frac{-2(x_{ij} - \mu)}{2\sigma^2} + \sum_{k \in g_j(r)} \frac{\kappa_p 2(\mu - \mu_p)}{2\sigma^2} && = 0 \\
&\rightleftarrows \frac{2}{\sigma^2} \left(- \sum_{k \in g_j(r)} \sum_{i=1}^{N} \tau_{ki} x_{ij} - \tau_{ki} \mu \sum_{k \in g_j(r)} \kappa_p \mu - \kappa_p \mu_p \right) && = 0 \\
&\rightleftarrows - \sum_{k \in g_j(r)} \sum_{i=1}^{N} \tau_{ki} x_{ij} + \mu \sum_{k \in g_j(r)} \sum_{i=1}^{N} \tau_{ki} + Z_j \kappa_p \mu - Z_j \kappa_p \mu_p && = 0 \qquad (3.17)
\end{aligned}$$

Solving Eq. (3.17) for μ yields the MAP estimator

$$\hat{\mu} = \frac{\sum_{k \in g_j(r)} \sum_{i=1}^{N} \tau_{ki} x_{ij} + Z_j (\kappa_p \mu_p)}{\sum_{k \in g_j(r)} \sum_{i=1}^{N} \tau_{ki} + Z_j \kappa_p}. \qquad (3.18)$$

The estimator for σ^2 is obtained analogously. From [53] the derivative with respect to σ can be written as

$$\begin{aligned}
\frac{\delta Q}{\delta \sigma} &= \frac{\sum_{k \in g_j(r)} \sum_{i=1}^{N} \tau_{ki} + Z_j(\nu_p + 3)}{\sigma} - \\
&\frac{1}{\sigma^3} \left[\sum_{k \in g_j(r)} \sum_{i=1}^{N} \tau_{ki} (x_{ij} - \hat{\mu})^2 + Z_j (\varsigma_p^2 + \frac{\kappa_p n_k}{\kappa_p + n_k} (\hat{\mu} - \mu_p)^2 \right] = 0
\end{aligned}$$

Solving for σ^2 yields the estimator

$$\hat{\sigma}^2 = \frac{\sum_{k \in g_j(r)} \sum_{i=1}^{N} \tau_{ki} (x_{ij} - \bar{\mu})^2 + Z_j (\varsigma_p^2 + \frac{\kappa_p n_k}{\kappa_p + n_k} (\bar{\mu} - \mu_p)^2)}{\sum_{k \in g_j(r)} \sum_{i=1}^{N} \tau_{ki} + Z_j (\nu_p + 3)}, \qquad (3.19)$$

again with $n_k = \sum_{i=1}^{N} \tau_{ki}$ and $\bar{\mu}_{kj}$ given by the ML estimator for μ Eq. (2.23) for structure g_{jr}.

3.6 CSI Mixtures and Clustering

Now that the MAP estimators for CSI mixtures have been derived, we will discuss some implications of the CSI formalism for practical data analysis and clustering.

3.6.1 Interpretation of the CSI Structure

One advantage of the CSI extension for practical data analysis is the high-level overview of the regularities found in the data. In order to illustrate this, consider the example structure in Fig. 3.5. In this figure the structure is color coded. Within each column of the matrix the same color indicates membership in the same group of the CSI structure. There are a

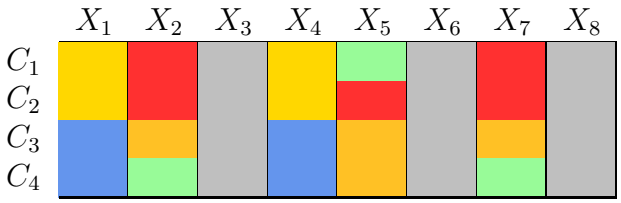


Figure 3.5: *Example of a color coded CSI matrix for four components and eight features.*

number of observations about the regularities which characterize the different components in this model which can read directly from this structure matrix

- Features X_3, X_6 and X_8 are not informative for the clustering.
- The four components fall into two general categories (C_1, C_2) and (C_3, C_4). Features X_1 and X_4 discriminate these general categories.
- C_1 and C_2 are subdivided into separate components by feature X_5 and the same is true for features X_2 and X_7 for components C_3 and C_4.

It should be stressed that this information is a direct outcome of the unsupervised learning procedure (see section 3.5) and does not require any human intervention. This kind of overview over the characteristics of the model is especially useful for data sets with many features. There the detailed analysis of the regularities characterizing a cluster requires considerable effort.

3.6.2 Feature Ranking

One typical problem during data analysis is to find which features contribute the most to the discrimination of the components for a given model. While the CSI structure explicitly captures features with do not contribute at all, a finer, quantitative ordering of the most informative features is often useful. Such an ordering can be obtained by ranking the features with an entropy-based score on the model parameters.

Probably the most common form of ranking would be to assign low ranks to features which capture a high degree of variability between components. One way to formalize that would be by the weighted symmetric Kullback-Leibler divergence KL of the parameters for a given feature, i.e.

$$\text{Score}(j) = \sum_{k_1=1}^{K} \sum_{k_2=k_1+1}^{K} (\pi_{k_1} + \pi_{k_2}) \text{KL}(\theta_{X_j|g_j(k_1)}, \theta_{X_j|g_j(k_2)}). \tag{3.20}$$

This score will assign large values to features which strongly discriminate the components. A ranking of features based on these scores in descending order will identify the most informative features.

An alternative, more refined form of ranking would be to find features which characterize a component (or a subset of components) against all other components. Such a ranking can be obtained as follows. In order to quantify the relevance of a feature X_j for component subgroup L, we assume a CSI structure in which X_j is discriminative for components L versus the other components, i.e. $Z_j = 2$ with $g_{j1} = L$ and $g_{j2} = \{1, ..., K\} \setminus L$. Based on this structure a component-specific parameter set θ_L and a parameter set for all other components θ_{other} are constructed by doing a single sEM update (see section 3.5.3).

The score for feature X_j and component in L is then given by

$$\text{Score}_{L,j} = \text{KL}(\theta_L, \theta_{\text{other}}), \tag{3.21}$$

where KL is again the symmetric relative entropy.

Chapter 4

Structure Learning Algorithm

In order to make practical use the advantages of the CSI formalism for data analysis, a reliable way to estimate CSI structures from data is required. In this section we give details for the CSI structure learning algorithm, discuss the combinatorial complexity and give results for several strategies for running time improvement.

4.1 Algorithm Overview

The basic outline of the CSI structure learning algorithm for a given data set D is as follows

Step 1 Run parametric EM (section 2.2) to obtain model M.

Step 2 Perform structural EM on M to obtain CSI structure.

The second step involves the scoring of possible CSI structures by their model posterior. Several exhaustive and greedy strategies for searching the structure space are discussed in section 4.3.

4.2 Combinatorial Complexity

A naïve approach to the CSI structure learning problem would require the scoring of each possible structure by its model posterior. The space of possible structures for a single feature is identical to all possible partitions of the component indices set $1, ..., K$ and increases exponentially with K. The exact number of possible structures can be computed by the Bell numbers B_K [2]. As an example for a single feature and ten components the number of possible structures to be evaluated is given by $B_{10} = 115,975$. For several features the number of possible combinations is then also exponential in the number of features p. For a given component number K and number of features p this means B_K^p possible structures in total.

Therefore, exhaustive enumeration of all possible structures is infeasible in for most real-world data sets and non-exhaustive search strategies over the structure space are required.

4.3 Structure Space Search Strategies

We considered the following search strategies over the space of possible CSI structures:

Full enumeration of all possible structures. This is only feasible for very small data sets but it yields an useful benchmark for the performance of the other strategies. As mentioned, the number of structures is B_K^p, exponential in both K and p.

Feature-wise enumeration of the possible structures. The structure is learned for each feature X_j separately and in each feature all possible structures are considered. The number of structures is $B_K p$, only exponential in K

Greedy, top down search For each feature the search is initialized with the full structure matrix (i.e. all components have a unique distribution). In an iterative fashion all pairwise merges of groups in the current structure are scored and the one which yields the best model posterior is retained for the next iteration. The maximal number of structures the greedy procedure will score is $O(K^3 p)$, i.e. cubic in K.

Greedy, bottom up search Similarly to the top down procedure, except that the initial structure is the case where all components are in the same group. All possible splits are scored to find the structure for the next iteration. The complexity of the search space is $O(K^2 p)$, i.e. quadratic in K.

The feature-wise, greedy learning of the CSI structure will converge to the global optimum of the model posterior in those cases where the optimal local structure g_j is identical to the structure of the feature on the globally optimal structure. To put it differently, if the globally best structure includes a locally suboptimal structure, the feature-wise procedures will not return the global optimum.

Exhaustive structure enumeration will be infeasible for most real world data sets. This means one of the greedy strategies will have to be applied. Therefore it is important to evaluate and quantify the difference in performance with respect to the global optimum of the model posterior as obtained by the complete enumeration (see section 4.3.2).

4.3.1 Choosing the Structure Prior

The first step of the structure learning procedure for a given data set is the choice of the hyperparameters γ and ω in the structure prior $P(M)$ (Eq. (3.5))

$$P(M) \propto P(K)P(G).$$

In general $P(M)$ encodes the preference for a simpler model. This is contrasted in the

model posterior $P(\Theta, M|D)$ with the data likelihood $P(D|\Theta, M)$ (Eq. (2.34)), which increases with model complexity. One way of thinking about the relation between prior and likelihood is that the prior acts as a regularizer of the likelihood to prevent overfitting by including too many parameters in the model. From the perspective of the CSI structure learning task, the choice of the hyper parameter ω of the structure prior $P(G)$ expresses the preference for a simpler, less complex structure. One way of looking at this, is that ω puts a threshold on the decrease in likelihood that is acceptable in exchange for a less complex structure. Since the likelihood of a data set is dependent on the sample size N the same must be true for ω. To make this explicit, consider the decision rule between a model M^0 with MAP parameters $\hat{\Theta}^0$ and a candidate model $M, \hat{\Theta}$ during an iteration of the learning algorithm. Assume that M^0 and M are identical except for a single merge in a g_j. This merge is accepted if

$$\frac{P(\hat{\Theta}^0, M^0|D)}{P(\hat{\Theta}, M|D)} = \frac{P(D|\hat{\Theta}^0, M^0)P(\hat{\Theta}^0, M^0)}{P(D|\hat{\Theta}, M)P(\hat{\Theta}, M)} \leq 1.$$

Under the assumption of uniform parameter priors, by substituting Eq. (2.34) and Eq. (3.5) and canceling terms this equals

$$\prod_{i=1}^{N} \frac{P(x_i|\hat{\Theta}^0)}{P(x_i|\hat{\Theta})} \ \omega \ \leq 1.$$

Each of the N fractions gives the decrease in likelihood of a x_i for moving from M^0 to the less complex model M. That is, we can think of each fraction as $(1 + \delta_i)$ where δ_i is the relative decrease in likelihood for x_i. Under the simplifying assumption that all of the δ_i are equal, i.e. $\delta_i = \delta$, we can now choose a δ as the *maximal relative decrease* in likelihood we are willing to accept in exchange for a less complex model. Then ω is given by

$$\omega = \omega(\delta, N) = \frac{1}{(1+\delta)^N}\,. \tag{4.1}$$

It is important to stress that at this point all we have done is to replace the choice of ω with the choice of δ. However this is advantageous for two reasons: First, the formula given above explicitly shows the impact of the data set size N. Secondly, δ has a straightforward interpretation based on the difference in likelihood between two models. As such it is easier to make an informed choice for δ based on the specific application.

The choice of γ is less crucial for the learning procedure. γ parameterizes the prior over the number of components $P(K)$. This prior has an impact on the posterior of models with different numbers of components. The effective number of components in a model changes if two or more components are in the same group for all features. The value of γ can be seen as an additional bias for a smaller number of components in the model posterior. This is useful for applications where the number of resulting clusters should be more strictly penalized. Otherwise γ can be chosen as the neutral, uniform prior with $\gamma = 1$.

4.3.2 Search Strategy Evaluation

In order to assess the performance of the different strategies, we randomly created models of different sizes and compositions. We considered models with Gaussian, discrete and both Gaussian and discrete features. The number of components varied by $K \in (2, 3, 4)$ and the number of features by $p \in (2, 3, 4, 5)$. For each model a CSI structure was chosen uniformly and model parameters were sampled according to the structure (see appendix D for details on random model generation). A data set of size 3000 was sampled from each generating model and MAP estimation of a conventional mixture of the appropriate size was performed to obtain the root model. The experiment was repeated 4800 times for each of the three data types (discrete, Gaussian, discrete & Gaussian). The different structure learning approaches were applied on different copies of this root model and the quality of the resulting structures was assessed. Here, the model posterior obtained with the exhaustive enumeration was used as benchmark, as this procedure yields the global maximum of $P(M|D)$ for a given data set and root model.

	All	Gauss	discrete	Gauss & discrete
Feature-wise enum.	99.29 %	99.75 %	99.01 %	99.08 %
Top down	99.26 %	99.75 %	98.99 %	99.02 %
Bottom Up	65.57 %	58.52 %	72.85 %	65.88 %

Table 4.1: *Global optimum obtained for non-exhaustive search strategies. It can be seen that the feature-wise enumeration and top-down local searches perform as well as the full enumeration in most cases.*

The results of the comparison with the full structure enumeration for the different types of generating models are shown in Tab 4.1. It can be seen that for all data types both the feature-wise enumeration as well as the top down procedure attained the globally optimal structure in almost all cases. These results indicate that the vast reduction in problem complexity that is obtained by fixing the feature order during structure learning does not carry too heavy a price with regards to the quality of the learned structures. The results for the different types of generating models are fairly consistent, with the exception of the bottom-up procedure which performed better for Gaussian than for discrete generating models. As one would expect, the results for heterogeneous generating models with both discrete and Gaussian features were in between the two pure settings.

While generally the local search strategies perform well in the experiments, the cases where the local procedures did not return the globally optimal structure warrant closer examination. The question being which factors are common to those models where the local search strategies did not return the global optimum. Two factors were identified which jointly characterize the cases where the local searches diverged from the global optimum. The first was the observation that most of the problematic generating models had redundant components in their structure, i.e. there were several components which shared the same group in all features. Fig. 4.1 shows an example of such a structure where components C_2

and C_3 share parameters in all features and could therefore be merged into a single component with weight $(\pi_2 + \pi_3)$. Such a structure introduces dependencies between the features

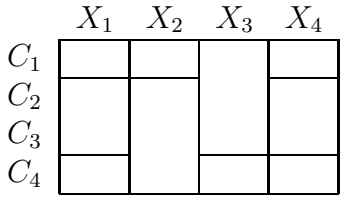


Figure 4.1: *Example CSI matrix with redundant components. Components C_2 and C_3 share parameters for all features.*

in the sense that the two components can only be merged to form the optimal structure, if all feature-wise structure searches also find this grouping. It does make sense then, that in such a situation the exhaustive enumeration of all possible structures will find the optimal structure, where a local search might miss it due to deviations of the trained parameters even for a single feature. The latter aspect leads us to the second characteristic of the problematic cases. It could be seen that component redundancy alone was not sufficient to cause divergent results, rather the parameter estimates of the parametric EM runs (Step 1 in section 4.1) had to be to some extent divergent from the true generating parameters. A typical case of such estimates would be that some component obtained a very small weight in the trained model whereas no such component existed in the generating model. In such a case one can say the parametric EM failed to sufficiently capture the generating model.

In summary, there seem to be two constellations were the local search is sub-optimal

- The generating model has redundant components and the parameter EM failed.
- The parametric EM failed by a wide margin.

The second case is fairly rare in comparison to the first. What both cases have in common is that they arise from the parametric EM taking a, relatively speaking, bad local maximum of the likelihood. This is a well known drawback of the EM procedure and it highlights the importance of using the algorithm in a manner which minimizes occurrence of such bad parameters (see section 2.2.4 for details). It also shows, that the exhaustive enumeration is able to compensate to some degree for an suboptimal parametric EM run by the sampling of the parameter space inherent to the structure search.

The next question was how the three local search strategies performed in comparison with each other. Tab. 4.2 shows the counts of experiments where one local search outperformed another, as measured by the model posterior. Each entry in the table gives the number of cases where the search strategy in the row outperformed the strategy for the column. In example, there were 4777 out of the 14400 cases where the feature-wise enumeration outperformed the bottom up approach.

It can be seen that the feature-wise enumeration and the top down search perform strongly consistent with only 8 cases where different structures were obtained (6 where the feature-

	Feature-wise enum.	Top down	Bottom up
Feature-wise enum.	-	6	4777
Top down	2	-	4778
Bottom Up	5	4	-

Table 4.2: *Performance comparison for the local search strategies. Each entry is the number of cases where the strategy in a row outperformed the strategy in a column.*

wise numeration was better, 2 where top down was). The bottom up procedure is commonly outperformed by the two other approaches. The main reason for that is probably the overly restricted search space of the split-based search, which only covers structures which fall in the default table representation. This problem could of course be addressed by an improved version of the bottom-up approach which includes both split and merge moves in the structure. However, considering the favorable results of the top down approach when compared to full local enumeration (which is optimal for fixed feature order), there is no need for a more complicated greedy search. Another interesting aspect to consider is the case where a locally suboptimal choice of structure leads to a better global structure. This situation is the only case where either the top-down or bottom-up search can outperform the feature-wise enumeration. By taking the sum over the feature-wise enumeration column in Tab.4.2 we get the total number of occurrences of such models. Out of the 14400 experiments run, only 7 such cases were observed. In these 7 cases the restriction of the structure search space inherent to the greedy procedures by chance led to a local structure which resulted in a global structure which was better than the feature-wise enumeration. Fortunately this this problematic constellation seems to occurs rather rarely.

Given all these results, the greedy top-down procedure stands out as the search strategy of choice. It combines computational efficiency with a strong performance in the structure learning. The results presented in the application chapters 5, 6 and 7 are based on the application of the top-down greedy search.

4.4 Running Time Optimization

In order to make structure learning feasible for larger data sets in acceptable running time, an efficient formulation of the learning algorithm and the top-down structure search is crucial. This includes caching strategies to avoid re-computation of certain terms and bounds on the model posterior to speed up the model posterior evaluation.

4.4.1 Feature-wise Caching

One useful consequence of the restriction to feature-wise search strategies is that the decomposition of the mixture likelihood can be used to speed up computations. From Eq. (2.1) and Eq. (2.2) we have the mixture likelihood

$$P(x_i|\Theta) = \sum_{k=1}^{K} \pi_k \prod_{j=1}^{p} P(x_{ij}|\theta_{kj}).$$

Assuming that we are currently learning the structure for feature X_{j^*} this can be written as

$$P(x_i|\Theta) = \sum_{k=1}^{K} \pi_k \left(P(x_{ij^*}|\theta_{kj^*}) \prod_{\substack{j=1,\\ j\neq j^*}}^{p} P(x_{ij}|\theta_{kj}) \right).$$

Now, all the terms in the product are equal for all structures of feature X_{j^*} and need only be computed once. This straightforward caching already decreases the running time of the structure learning by a factor of $O(p)$.

4.4.2 Candidate Structure Graph

The top-down search strategy considers in each step all pairwise merges of groups in the current structure and accepts the merge which yields the highest model posterior. An example of this is shown in Fig. 4.2a). For clarity we denote $\theta_{X_j|g_{jr}}$ with the simplified notation $\theta_{g_{jr}}$ in the example. Each of the four nodes in the figure represents a component of the mixture and each pair of components gives rise to a merge parameter $\theta_{g_{jr}}$ based on the expected sufficient statistics of the merge (see section 3.5.3), which in turn allows the evaluation of the model posterior $P(\hat{\Theta}, M|D)$.

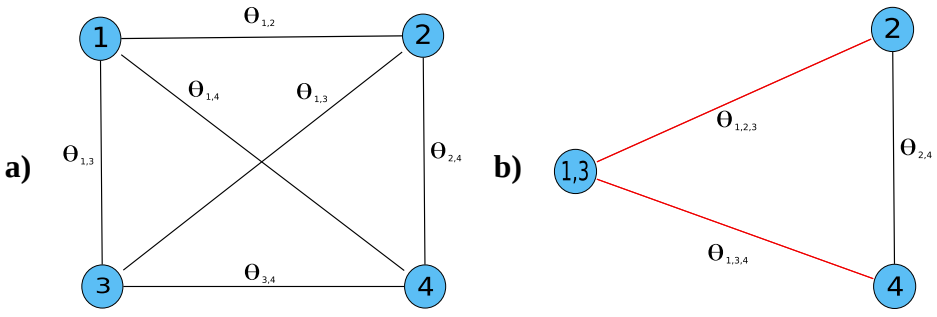


Figure 4.2: ***a)*** *Pair-wise merges to be evaluated in the first step of the greedy structure learning for a four component mixture.* ***b)*** *Second step after $\theta_{1,3}$ has been accepted in a). Only the parameters corresponding to the red edges need to be recomputed.*

This means that in each step $O(Z_j^2)$ candidate merges have to be computed, where Z_j is the current number of groups, starting with $Z_j = K$ in the first step. An important observation

that can be made, is that the merge parameters $\theta_{g_{jr}}$ of disjunct merges are independent in the sense that the respective computations have no terms in common. This is because the merge parameters are computed from the element-wise addition of the component membership posteriors $\tau_k = \{\tau_{ki}\}_{i=1,\ldots,N}$ of the components that are part of the merge (see Eq. (3.15)). An example would be $\theta_{1,3}$ and $\theta_{2,4}$ in Fig. 4.2a). The former is based on $\tau_{1,3} = \tau_1 + \tau_3$, whereas the latter arises from $\tau_{2,4} = \tau_2 + \tau_4$. If we were to accept the merge of 1 and 3 in the first step, the second step (shown in Fig. 4.2b)) would necessitate the re-computation of only the merge parameters $\theta_{1,2,3}$ and $\theta_{1,3,4}$ (edges shown in red), whereas $\theta_{2,4}$ would remain unchanged from the previous step and need not be computed again. Therefore, by caching the merge parameters in each step, the complexity of merge parameters to be re-evaluated in each step after the first drops from $O(Z_j^2)$ to $O(Z_j)$. This greatly increases the speed of the structure learning, especially for models with a large number of components.

4.4.3 Posterior bounds

Another approach for speeding up the evaluation of candidate structure model posteriors is to derive upper and lower bounds on the candidate structure decision function. This decision function is simply the fraction of model posteriors for the current model M^0 and the candidate model M^1, i.e.

$$\text{Dec} = \frac{P(\hat{\Theta}^0, M^0|D)}{P(\hat{\Theta}^1, M^1|D)} < 1.$$

Given analytical lower bounds Dec^{low} and upper bounds Dec^{up} on Dec which can be computed efficiently we could make use of $\text{Dec}^{up} < 1 \Rightarrow \text{Dec} < 1$ and $\text{Dec}^{low} > 1 \Rightarrow \text{Dec} > 1$ to obtain a faster decision for a given candidate merge. The significant caveat to this is that the bounds need to be sharp with respect to Dec, in order for a decision to be possible. Since $\text{Dec}^{up} > 1 \not\Rightarrow \text{Dec} > 1$ (and respectively for Dec^{low}) an insufficiently sharp bound will not resolve the decision and Dec has to be evaluated exactly. Note also that Dec^{up} is the more useful bound as it allows for the quick identification of structures which can be discarded. The developments in this section are independent of the CSI extension. Therefore the somewhat easier formulation of Θ from the conventional mixtures will be used in the remainder of the section.

In the following we give examples for possible definitions of Dec^{low} and Dec^{up}. The first is a simplistic example, which illustrates the principle. The second can be seen on an extension of the first making use of properties of the logarithm.

Substituting Eq. (2.34) into Dec yields

$$\text{Dec} = \frac{\prod_{i=1}^{N} P(x_i|\Theta^0) P(\hat{\Theta}^0, M^0)}{\prod_{i=1}^{N} P(x_i|\Theta^1) P(\hat{\Theta}^1, M^1)}$$

$$= \prod_{i=1}^{N} \frac{\sum_{k=1}^{K} \prod_{j=1}^{p} \pi_k P(x_{ij}|\theta_{ij}^0)}{\sum_{k=1}^{K} \prod_{j=1}^{p} \pi_k P(x_{ij}|\theta_{ij}^1)} \frac{P(M^0)}{P(M^1)}.$$

Assuming j^* is currently being learned we let $L_ik = \prod_{j=1, j\neq j^*}^{p} \pi_k P(x_{ij}|\theta_{ij})$ and $l_ik = \pi_k P(x_{ij^*}|\theta_{ij^*})$ which yields

$$\text{Dec} = \prod_{i=1}^{N} \frac{\sum_{k=1}^{K} L_{ik}^0 l_{ik}^0}{\sum_{k=1}^{K} L_{ik}^1 l_{ik}^1} \frac{P(\hat{\Theta}^0, M^0)}{P(\hat{\Theta}^1, M^1)}.$$

Now define $l_i^0 = (l_{i1}^0, ..., l_{iK}^0)$ and analogously l_i^0. It should be noted that in fraction of the two priors, almost all terms cancel and it is therefore trivially to evaluate.

Simplistic bounds

Given that all elements in l_i^0, l_i^1 and L_{ik} are positive, and that the L_{ik} are identical in numerator and denominator, bounds Dec^{low} and Dec^{up} can be obtained as

$$\text{Dec}^{low} = \prod_{i=1}^{N} \frac{\min(l_i)}{\max(l_i)} \frac{P(\hat{\Theta}^0, M^0)}{P(\hat{\Theta}^1, M^1)}$$

and

$$\text{Dec}^{up} = \prod_{i=1}^{N} \frac{\max(l_i)}{\min(l_i)} \frac{P(\hat{\Theta}^0, M^0)}{P(\hat{\Theta}^1, M^1)}$$

These bounds illustrate the principle. In practice however they are rarely sharp enough to allow a decision based on Dec^{low} and Dec^{up} alone.

Logarithmic bounds

Since numerator in Dec is the model posterior $P(\hat{\Theta}^0, M^0|D)$ of the currently best structure found, we can consider it to be constant and known. In this case we only need bounds for the likelihood function of the candidate structure $P(D|\hat{\Theta}^1, M^1)$ (as the prior is unproblematic). This likelihood is given by

$$P(D|\hat{\Theta}^1, M^1) = \prod_{i=1}^{N} P(x_i|\Theta^1),$$

often it is more convenient to use the $\log$–scale, which is also consistent with the actual

implementation due to underflow avoidance. The $\log$–likelihood is

$$\log P(D|\hat{\Theta}^1, M^1) = \sum_{i=1}^{N} \log \sum_{k=1}^{K} L_{ik} l_{ik},^1$$

as the function to be bounded. Now, the $\log$–scale computations yield the $\log$ values of $L_k l_k, k = 1, ..., K$, which means that the $\log$ of the inner sum cannot be straightforwardly evaluated. We apply the standard solution in form of the $sumlogs$ function [121] to compute $\log \sum_{k=1}^{K} L_{ik} l_{ik}$ based on $\log L_{ik} l_{ik}, k = 1, ..., K$. One property of the $sumlogs$ function is that is requires the maximum over the $\log L_{ik} l_{ik}$ as an input. This implies that first all $\log L_{ik} l_{ik}$ have to be computed before $sumlogs$ can be applied. The evaluation of $\log \sum_{k=1}^{K} L_{ik} l_{ik}$ requires then two $O(NK)$ passes over the data and mixture components.

The bounds we are about to propose are based on the observation that the structure learning operates on the MAP parameters $\hat{\Theta}$, this implies that for most data points typically one component k^* will yield a much higher value than the others. The exception here are only data points which lie at the boundary of component densities or within strongly overlapping components. It is a property of the logarithm that then $L_{ik^*} l_{ik^*}$ will dominate in $\log \sum_{k=1}^{K} L_{ik} l_{ik}$. This fact can be made use of to define a lower bound $P^{low}(D|M^1)$ as

$$P^{low}(D|\hat{\Theta}^1, M^1) = \sum_{i=1}^{N} \max_{k=1,...,K} \log L_{ik} l_{ik}.$$

That is, all but the largest element of the inner sum are omitted. Since we are dealing with non-negative values, this yields a lower bound on the sum and the logarithmic properties ensure that the bound will be sharp in many cases. For the upper bound we need to define the function scdmax which simple returns the second largest element of a vector. Then

$$P^{up}(D|\hat{\Theta}^1, M^1) = \sum_{i=1}^{N} \max_{k=1,...,K} \log L_{ik} l_{ik} + ((K-1) \operatorname*{scdmax}_{k=1,...,K} \log L_{ik} l_{ik})$$

gives the upper bound. Here each of the non-maximal summands has been replaced by the second largest $\log L_{ik} l_{ik}$. Since typically the maximal term in the sum is much larger than the second largest, this will give a tight upper bound of $\log P(D|\hat{\Theta}^1, M^1)$.

The computational advantage of these bounds lie in that the largest and second largest value of $\log L_{ik} l_{ik}$ can be propagated with very little effort while they are computed. If the bounds allow for a quick rejection of a structure, the evaluation of $sumlogs$ is omitted.It should be noted that these bounds do not reduce the big-O complexity of the algorithm but can yield improved running times in practice (see following section).

[1]Note that the $L^1_{ik} l^1_{ik} = L_{ik} l_{ik}$ for the rest of this section

4.4.4 Structure Learning Running Time

We compared running times of the basic structure learning, structure learning with cached candidate history as described in section 4.4.2 and candidate history plus logarithmic bounds (see previous section) for three different model types. The types were Gaussian mixtures, discrete mixtures and mixtures with both discrete and continuous features. The data set sizes N used were 1000, 3000 and 10000. The number of components were K 10, 15, 20 and 25. Each model had 40 features with randomly selected parameters and CSI structure (see appendix D for details of random model generation). Boxplots for the running times of 100 repetitions of the experiment for the discrete and Gaussian variants are given in Fig. 4.3 and Fig. 4.4 respectively.

First consider the results for Gaussian mixtures in Fig. 4.3. It can be seen that for ten components all variants perform similarly for all values of N. For higher component numbers, the structure history yielded a smaller improvement in running time, while the improvement conveyed by the bounds increases. For the discrete mixtures (Fig. 4.4) we find consistently that the structure history considerably improves the running time behavior, whereas the addition of bounds actually has a detrimental effect. The results for the setup with both discrete and Gaussian features again produced results in between the two pure cases (not shown).

The reason that the learning history did not yield a great improvement for the Gaussian models when compared to the discrete mixtures lies in the different numbers of degrees of freedom for these distributions. The learning history reduces the number of necessary computations of MAP parameter estimates (section 2.4.2). While the Gaussian distribution only required estimates for the mean and variance parameters, the discrete distribution used were defined over an alphabet of size eight, making the MAP estimates more expensive, and conversely the learning history more useful.

Another interesting aspect of these results is that the bounds actually had a detrimental effect on running time for the discrete models. The reason for that is due to an inherent property of the distributions, namely how clearly the component parameters of a model are separated. One typical way to quantify this separation is the relative entropy. For discrete distribution the relative entropy is bounded by $\log_2 M$ whereas for continuous distributions the entropy is unbounded. Since the sharpness of the bounds increases with the separation of the components, the bounding scheme is more likely to bear fruit for Gaussian distributions.

Based on these results the use of the bounds should be reserved for the use on continuous data, whereas the learning history is of general use.

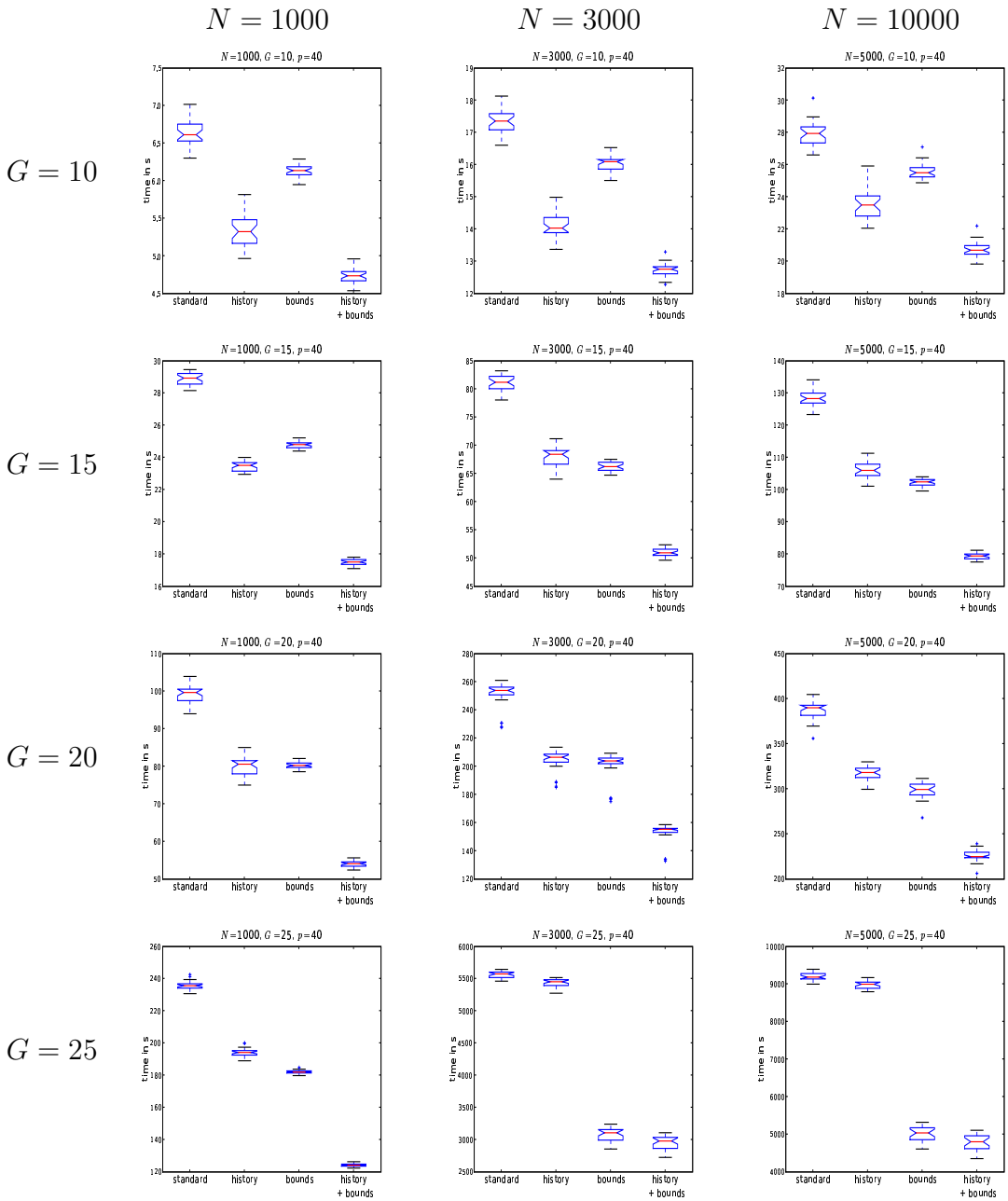


Figure 4.3: *Running time comparison plots for Gauss models with* $G \in (10, 15, 20, 25)$, $N \in (1000, 3000, 10000)$ *and p = 40*

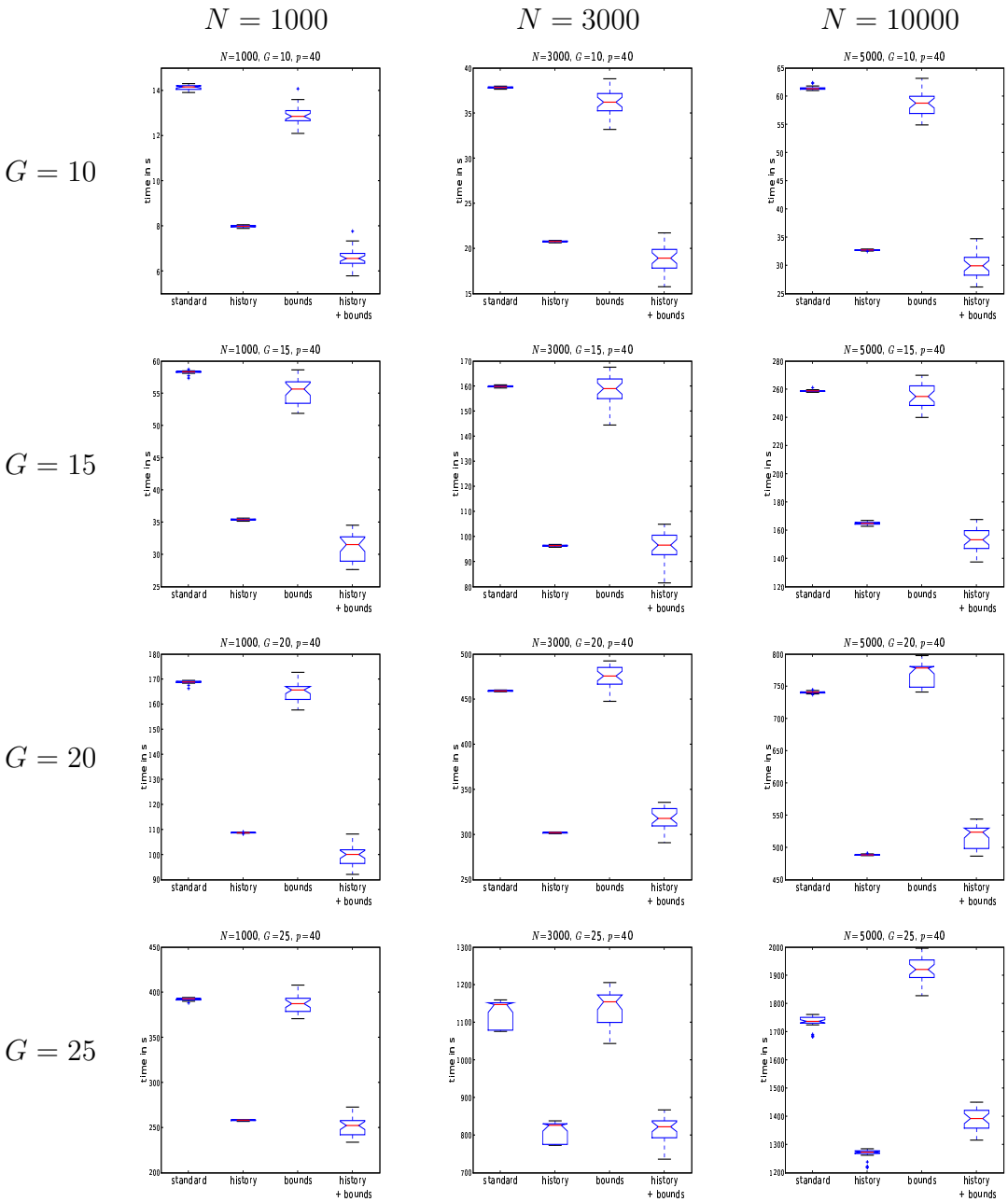


Figure 4.4: *Running time comparison plots for discrete models with* $G \in (10, 15, 20, 25)$, $N \in (1000, 3000, 10000)$ *and p = 40*

Chapter 5

Mixture Modeling for Transcription Factor Binding Sites

In this chapter we describe the application of CSI mixtures for the modeling of transcription factor binding sites (TFBS). In the following section 5.1 the we provide some biological background on the problem setting. In section 5.3 we evaluate the performance of the CSI mixtures on both simulated and biological data.

5.1 Introduction

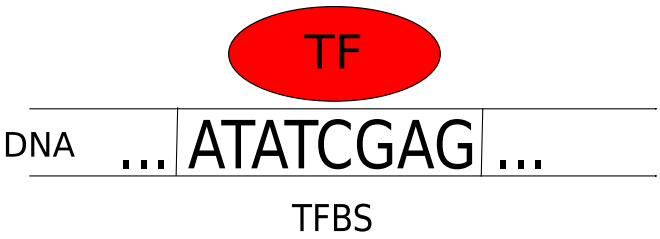


Figure 5.1: *A transcription factor binds to a specific stretch of DNA in the genome.*

The binding of transcription factors (TF) to specific stretches of genomic DNA is one of the major mechanism of gene regulation. Fig. 5.1 shows a cartoon of such a binding event. Through chemical interactions of the TF and the genomic DNA, TFs bind to specific stretches on the genome, the binding sites. These binding sites are specific for each factor and the study of the binding behavior of TFs is a problem of considerable importance for understanding gene regulation. The accepted approach is to formulate a mathematical representation of the binding pattern of a given factor based on collections of confirmed binding site sequences. This representation is subsequently used to score candidate sequences for occurrences of said pattern. The effectiveness of this approach depends on the models ability to accurately formalize the regularities found in the confirmed sites. The most commonly taken approach is the positional weight matrix (PWM) model [175, 176, 179, 180, 194]. PWMs are a statistical approach to modeling the factor-specific binding site composition. A PWM is derived from a multiple alignment of con-

firmed binding sites. For each position in the alignment a distribution over the four bases is estimated from the corresponding alignment column. Assuming independence between positions, this gives a probabilistic model of the binding site of a specific factor which subsequently can be used to score whether a DNA sequence contains a binding site for this factor [85, 107]. From the perspective of mixture models, a PWM is simply a single component naïve Bayes model (Eq. (2.1)).

However, the PWM approach relies on two strong assumptions, namely that *all* positions

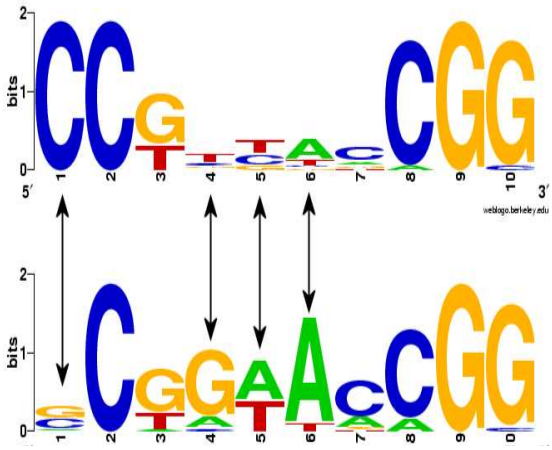


Figure 5.2: *WebLogos (*http://weblogo.berkeley.edu*) for the two subgroups of Leu3 binding sites. It can be seen that sequence variability is limited to positions 1, 4, 5 and 6 (indicated by arrows).*

within the site are independent and, more importantly, that all binding sites of a factor are slight variations of the *same* sequence. The former has been shown to be a simplification of biological reality for such examples as the Zinc finger motive [199] or the Mnt repressor [120]. For the latter there is ample biological evidence to make it at least doubtful: It is well known that TFBS occur in clusters of functionally interacting TFs in promotor regions, so called transcriptional modules [20, 118, 185]. A single factor may have many different interaction partners for different genes and it has been shown that the topology of these modules has an impact on the binding site sequences found for about nine thousand sites in *S. cerevisiae* [18]. Also, it is known that a single change in a binding site can have profound effects on both the interaction behavior of a factor [148] or the level of induced gene expression [196]. Moreover, in [93] the authors find increased levels of conservation for non-consensus binding site positions for 16 factors in 10 bacterial genomes, concluding that these sites are subject to evolutionary pressure. Finally, in [188] the authors confirmed the presence of position dependencies within a set of binding sites taken from the JASPAR data base [158]. This gives further evidence for a level of biological complexity of binding site sequences beyond the “single site” hypothesis and motivates the development of more sophisticated methods.

This issue has received some attention in recent years. In [9] the authors successfully used subclasses of Bayesian networks for *de novo* motive discovery, among them mixtures of PWMs. More recently, in [79] binding sites have also been described as mixtures of

PWMs. There it was shown, that a two component mixture model yielded improved conservation scores and higher expression coherence when compared to using a single PWM for a collection of 64 PWMs taken from the JASPAR.

However, there are several drawbacks of the conventional mixture approach as it was introduced in chapter 2. Namely, the essentially unsolved problem of choosing an appropriate number of mixture components, in particular if data is sparse and the classical model selection techniques (see section 2.3.1) will not perform well. In general too few components lead to suboptimal performance due to insufficient generalization, while, more severely, too many components will cause overfitting. To circumvent this issue the number of components was fixed to two in [79]. Moreover, it seems plausible that for most factors which have several types of binding sites (and can thus be modeled more precisely by a mixture), the different subgroups will not consists of distinct, dissimilar sequences. Rather, the variability between sites will be concentrated on specific positions. Estimating a full PWM for each mixture component will then introduce unnecessary parameters into the model. This increases model complexity unnecessarily and leads to less robust parameter estimates.

These issues are addressed in the automatic adaptation of model complexity inherent to the CSI framework. Therefore CSI mixtures are a natural choice of model to capture the full biological signal of TFs with complex binding behavior.

In this context the CSI principle introduced in section 3.3 amounts to representing binding site positions with little variability in the different components by the same distribution. A biological example for such a situation is the TF Leu3. In [79] the authors showed that a two component mixture naturally separated the known binding sites [113] into one high and one low binding-energy subgroup. Now, consider Fig. 5.2. The figure shows the sequence logos [163] for these subgroups. It can be seen that sequence variability is only present in position 1, 4, 5 and 6 (indicated by arrows) while the other sites are highly conserved. Another example is the factor Reb1. Reb1 binds with different affinities to motives TTACCCG and TTACCCT [191], that is the two subgroups differ in a single position only.

As described in previous sections, the advantage of the CSI model in settings such as the Leu3 and Reb1 data is that in a conventional mixture random sequence deviations will cause the parameters in the different components for the same position to vary slightly, even if there is no biologically meaningful variability on the sequence level. This overfitting introduces a distortion in the scores produced by the model that may result in a decrease in performance. Therefore, learning a CSI structure does not only yield a more parsimonious model, as less parameters are required, but also increases robustness for noisy data.

In the following sections we will evaluate the performance of the CSI method for TFBS data based on both simulated and real biological data.

5.2 TFBS Modeling

Modeling Choices

Since transcription factor data consists of DNA sequences, the models are mixtures of discrete distributions over the four bases, i.e. $\Sigma = (A, C, G, T)$. The parameter prior is a product of conjugate Dirichlet priors (see section 2.4.1). The prior over the mixture weights π was uniform, the priors over the $\theta_{X_j|g_{jr}}$ were chosen to be almost uniform with a small bias towards uniform θ (i.e., all hyperparameters of the Dirichlets were set to 1.02). This was done to guard against overfitting by setting zero probabilities in the parameter estimation.

For this application it seemed reasonable to use a strong prior, such that the structure only introduced additional complexity into the model if clearly warranted by the data. In the following we choose hyperparameters for the prior according to the heuristic Eq. (4.1) with $\omega(0.18, N)$ (unless noted otherwise). As an example for 20 sequences we obtain $\omega(0.18, 20) = 0.036$.

Sequence Scoring

One practical advantage of the CSI model extensions is that it refines the models ability to represent TF binding patterns without abandoning the framework of probabilistic models. This means that the CSI model can be seamlessly and easily combined with established techniques for finding hits with significant scores in genomic sequences [86, 107]. Here, as in [79], the score of a mixture was defined as the maximum score over all components. This means that the score of a sequence was given by the strongest signal found among the components. Similar scoring schemes have been used for instance in the field of speech recognition.

5.3 Results

5.3.1 Simulation Studies

In order to examine the difference in performance between conventional mixture and CSI models we generated artificial data sets from mixtures with differing numbers of components and structures.

In the first experiment the generating model was a two component CSI mixture with $p = 10$ and random weights π. The CSI structure was set up as follows: Out of the ten positions, six were represented by single distributions in both components and four had a unique distribution in each component. The parameters of the distributions $\theta_{X_j|g_{jr}}$ were chosen randomly.

Generating model	Best trained model	avg. ΔBIC
G_1	M_1	32.36
G_2	M_2	70.28
G_{CSI}	M_{CSI}	232.16
G_4	M_4	149.31

Table 5.1: *Optimal model for the four data sets according to the average difference in BIC to the BIC of the generating model over 30 repetitions.*

First we evaluated the ability of our method to adapt to the structure in the data and thus to avoid overfitting. We trained one conventional and one CSI mixture model, both using three components on a training data set with 40 samples. The first result was that the structure learning algorithm recovered the generating models two component CSI structure with high accuracy (not shown). In order to quantify the advantage of the CSI model for sequence scoring we generated test data sets with 500 samples. We used a uniform background model to obtain the scores for each sample and the scores were then converted to p-values based on a score distribution on 1Mb of random sequence. We repeated the simulation for 30 different randomly generated data sets and observed that the CSI mixture yielded better (lower) p-values than the conventional mixture. The one-sided Wilcoxon test for paired samples assigned a significance of 0.02 to this result. Repeating the experiment with only 25 training samples confirmed these results with a Wilcoxon test significance of 0.04.

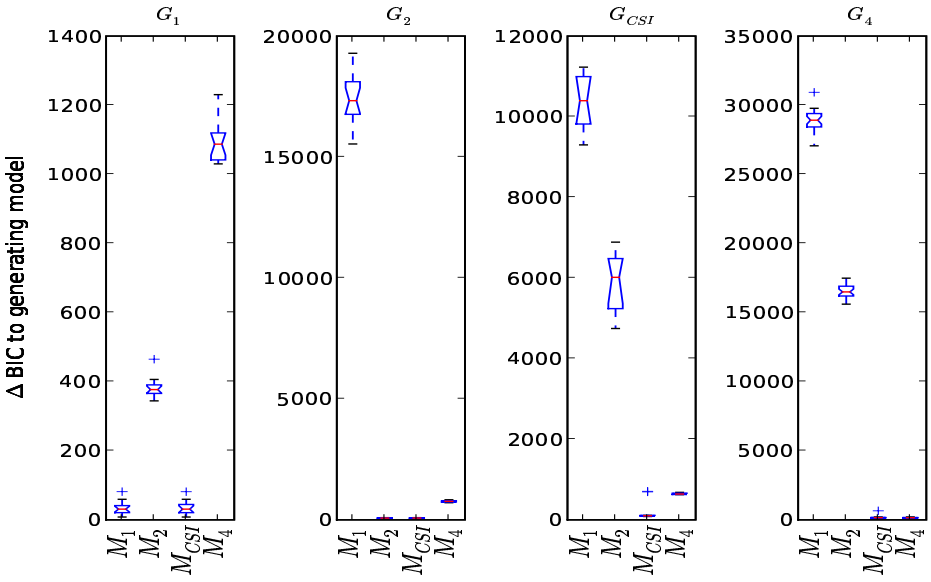


Figure 5.3: *Distributions of the difference in BIC to the generating model for the four simulated data sets on 30 repetitions.*

The next question we addressed was how the CSI model performed for different data sets in a classical model selection setup. We generated data sets of size 3000 with $p = 12$ from

four different models: a single PWM model G_1, a conventional two component mixture G_2, a CSI mixture with four components G_{CSI} and a conventional four component mixture G_4. The parameters of the discrete distributions in Θ were chosen such that one base β was assigned a random probability sampled uniformly from [0.6,0.8] and the remaining mass split randomly over the other bases. In each case β was chosen such that it adhered to the CSI structure of the respective model, that is components that did not share a group for a X_j also had a dissimilar β. The structure in G_{CSI} consisted of 4 positions with four groups, four positions with two groups and four positions with one group each. This means that the CSI model matched G_1, G_2 and G_4 in complexity for four features each.

Subsequently, we trained 30 models M of each of the four types (i.e. M_1, M_2, M_{CSI} and M_4) on each of the four generating model types. Model fit was assessed by BIC (see 2.27). As a base value the BIC of the generating model on the test data set was computed.

Table 5.1 shows the average difference in BIC of the best trained model (M_1, M_2, M_{CSI} or M_4) when compared to the BIC score of the true model. As one would expect, the model type that best matches the respective generating model yields the optimal BIC. A more interesting point to consider was the distributions of the differences of the BIC scores of the different models to the BIC obtained under the generating model shown in Fig. 5.3. It can be seen that over the range of generating models M_{CSI} achieves model selection scores comparable to those models which match the generating type. These results illustrate the inherent ability of CSI models to adapt to different data settings. This makes CSI a preferable choice of model for practical applications where the true number of components is unknown.

5.3.2 Analysis of TF LEU3

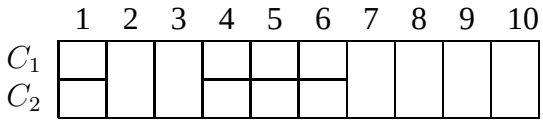


Figure 5.4: *Two component CSI mixture structure for known Leu3 binding sites. Each cell represents a discrete distribution, where cells spanning both rows identify positions with high conservation in both subgroups.*

It was shown that 46 known binding sites of the TF Leu3 [113] can be separated into a high and low binding-energy subgroup using a two component mixture with highly significant p-value [79]. We repeated this analysis by training a two component CSI mixture. Since we were using the model in a clustering context a weak prior of $\omega(0.05, 46) = 0.11$ was used. Fig. 5.4 shows the resulting CSI structure. Note the correspondence between the fully parameterized positions (1, 4, 5, 6) and the group specific sequence variability as visualized in Fig. 5.2. The CSI mixture yielded a subgroup division of the Leu3 sites that was practically identical to the one previously reported. However there are two important differences between the two models: First, the conventional mixture requires the estima-

tion of 61 free parameters while due to the tying expressed in the CSI structure our model only needs 43 parameters. This means that CSI gave equivalent results using about 30% less parameters. Secondly, the CSI structure makes information about the subgroup and position specific sequence variability an explicit part of the model. Having this information readily available will facilitate further investigations, especially for large-scale studies where hundreds or more factors are involved.

5.3.3 Conservation Statistics

The validation of predicted binding sites with respect to their biological functionality is a difficult problem as functionality cannot be assessed directly. One surrogate for functionality found in the literature is the degree of conservation in genomic sequences between related species [184]. For the sake of comparability with the results reported in [79] we follow the same evaluation approach taken there and evaluate the different models by the fraction of conserved predicted binding sites.

In the following we are going to evaluate the performance of a single PWM M_1, a two component mixture M_2 and a two component CSI mixture M_{CSI} based on human-mouse conservation. We used the same 64 JASPAR TFs as in [79]. We downloaded the 1kb upstream regions of the **hg17** assembly (May 2004) from the UCSC genome data base [87]. The mouse conservation data (**mm7**) was extracted from the axtNet data set [165] (also UCSC). For each of the 64 TFs and each of the three models under consideration, we then computed the 1000 best scoring hits in the 1kb upstream regions. The overall base composition of the sequences was used as the background model. For the mixtures the hits were chosen proportionally to the mixing weights. This means that for a $\pi = (0.6, 0.4)$ we would chose the 600 best hits from the first component and the 400 best from the second. The fraction of hits that was conserved in mouse was then computed based on a 80% sequence identity cutoff.

Evaluation: In order to decrease the impact of random variation on the analysis we considered TFs with very similar fractions of conserved hits for two model types as not giving conclusive preference to any of the two. That is, if the difference in the conserved fraction was less than ten percent of the maximal conserved fraction observed for any of the three model types, the scores were considered to be ”equal” for the purposes of this analysis. This has the effect of making the results more conservative in the sense that the impact of factors with very small differences in the conservation statistics was suppressed.

Fig. 5.5 shows the comparison of conserved fraction for the three model types. To illustrate the impact of the available number of training samples N for a factor on performance, we depict TFs differently based on the number of associated sequences. TFs with less than 18 sequences are shown as red diamonds, TFs with 19 - 31 sequences are shown as blue rectangles and TFs with more than 31 sequences are shown as green dots. The numbers

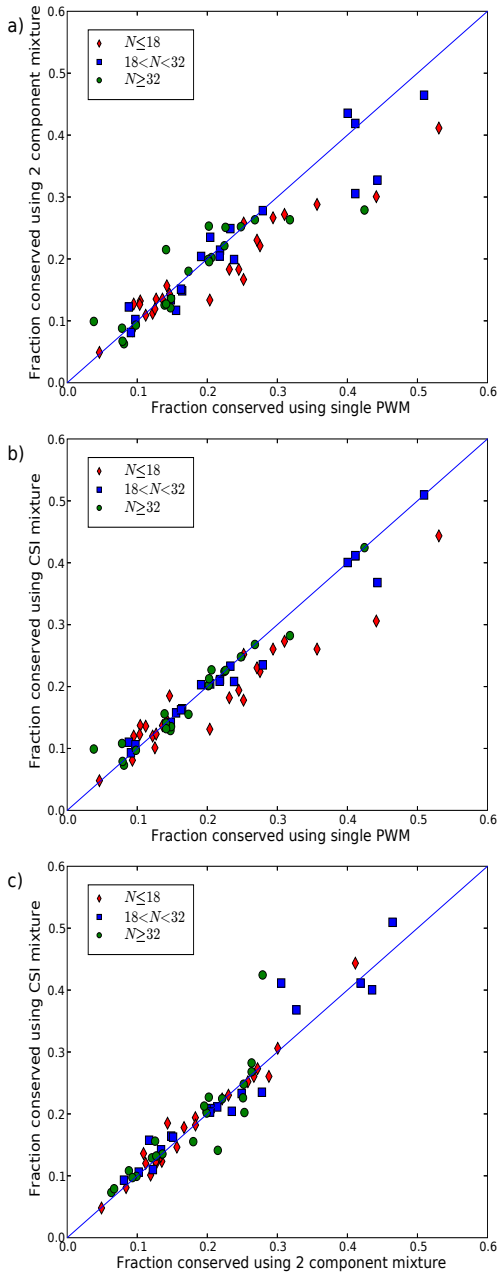


Figure 5.5: a) *Conserved fractions of hits for M_1 and M_2. The mixture M_2 is as good or better for 67% (43) of the TFs.* ***b)*** *Conserved fractions for M_{CSI} and M_1. For 70% (45) of the TFs the conservation of M_{CSI} was as good or better than for M_1. Outliers with strong preference for M_1 model had very few known sequences. If we only consider TFs with at least 20 sequences, the CSI yields as good or better conservation in 85% (34/40) of the cases.* ***c)*** *Comparison of conservation statistics of M_2 and M_{CSI}. For 89% (57) of the TFs M_{CSI} yields higher or equal conservation.*

	$M_2 \geq M_1$ (43)	$M_1 > M_2$ (21)
$M_{CSI} \geq M_2$	84% (36)	100% (21)
$M_{CSI} > M_2$	47% (20)	81% (17)
$M_{CSI} \geq M_1$	89% (38)	33% (7)
$M_{CSI} > M_1$	37% (16)	10% (2)

Table 5.2: *Comparison of the conserved fraction of the 1000 best scoring hits for M_{CSI}, M_1 and M_2 in the two subsets of the TF data given the conditions $(M_2 \geq M_1)$ and $(M_1 > M_2)$ respectively.*

were chosen as to split the 64 TFs into three roughly equally sized groups.

In the following we compare and contrast the results for the three different model types M_1, M_2 and M_{CSI}.

M_1 **vs** M_2: In 5.5a) you can see the conserved fraction of M_1 and M_2 for the 64 TFs in the data set. The mixture model M_2 was as good or better than M_1 in 67% (43) of the cases. For 33% (21) of the TFs the mixture was strictly better. This means that the performance of the two component mixture was somewhat weaker in our analysis than reported in [79] . Recall, that our data set differed from the one in [79] as it was based on a later genome freeze and, more importantly, it did not contain any downstream sequences. To the best of our knowledge the rest of our analysis was identical to the one conducted in [79].

M_{CSI} **vs** M_1: The comparison between the fraction of conserved hits of the CSI mixture M_{CSI} and the single PWM model M_1 can be seen in Fig. 5.5b). In 70% (45) of the TFs under consideration M_{CSI} showed a conserved fraction as good or better than M_1, with 28% (18) being strictly better. One important observation is that in most instances where M_1 had a strong advantage in conserved hits, the factor had only a small number of known binding sites. This can be seen by the large number of diamonds below the diagonal. For instance the rightmost point in Fig. 5.5b) at (0.53, 0.43) corresponds to MA0062 which has 7 known sites. In such a situation a little random variation in the sequences can have a strong impact on the trained model and lead to spurious structures. This is supported by the correlation between the number of available sequences for a factor and the increase in conservation for the CSI model. If we only considered TFs with 15 or more sequences, M_{CSI} is as good or better in 74% (40/54) of the cases, for 20 or more sequences in 85% (34/40) and for 40 or more in 94% (15/16). The fraction of TFs where M_{CSI} is strictly better remained in the range of 30% independent of the number of sequences.

M_{CSI} **vs** M_2: In Fig. 5.5c) we show the fraction of conserved hits for M_{CSI} and the conventional two component mixture M_2. For 89% (57) of the TFs the CSI model yields higher or equal conservation, 58% (37) being strictly greater.

Performance of M_{CSI}: Applying the two conditions $(M_2 \geq M_1)$ and $(M_1 > M_2)$ on the conserved fractions of hits split the 64 TFs in two subsets of size 43 and 21. We can think of the first subset as those TFs where a mixture model is appropriate and the second subset as being better represented by a single PWM. In the following we examined the performance of our CSI models within these two subsets. The results are summarized in Table 5.2. For the subset induced by $(M_2 \geq M_1)$ M_{CSI} was as good or better then M_1 or M_2 for a strong majority of 84% (36) and 89% (38) of the TFs respectively. M_{CSI} was strictly better for 47% and 37% respectively. This means that for TFs where a two component mixture improves performance as compared to a single PWM, the CSI model will in most cases outperform both of the other models. M_2 due to the reduction in overfitting and the more robust parameter estimates, M_1 because of the improved description of the binding pattern.

For the subset where a single PWM yielded a larger conserved fraction than the two component mixture (given by the condition $(M_1 > M_2)$) M_{CSI} was as good or better than M_2 for all the TFs in the subset (100% (19)) and strictly better for 81% (17). This illustrates the property of the CSI model to adapt to the number of subgroups supported by the data (one in this case) by means of the structure learning. M_{CSI} is equivalent or better than M_1 in 33% (7) of the TFs in the subset. This rather low number again shows the impact of spurious structures for TFs with few known binding sites. If we only consider the 11 TFs in the subset with 20 or more annotated binding sites, the value for $(M_{CSI} \geq M_1)$ goes up to 64% (7/11). Finally, M_{CSI} is strictly better than M_1 for a negligible 10% (2). This is not surprising as we would not expect CSI to outperform M_1 in situation where a single PWM is the appropriate model. Rather a successful application of the structure learning in such a case makes M_{CSI} equivalent to M_1. This corresponds to the points which lie directly on the diagonal (i.e. the conserved fractions are equal) in Fig. 5.5b).

5.3.4 Examples of Binding Site Subgroups

Out of the 64 TF under consideration 41 showed two groups in the CSI structure, for the remaining 23 the structure was completely merged into the single PWM case. In Fig. 5.7 we show the sequence logos of four examples of subgroup specific binding patterns and the corresponding CSI structure. The factors are Foxd3 (MA0041), HLF (MA0043), Foxa2 (MA0047) and CEBP (MA0102). The double arrows mark the positions where two distributions were taken in the learned CSI structure.

As one would expect, it can be seen that these positions correspond to the most strongly discriminatory positions between the two sequence logos. It is also interesting to note that the discriminatory positions are unevenly distributed in the examples. For Foxd3, for instance, these positions are concentrated at the beginning of the binding site whereas for Foxa2, they are evenly spread along the length of the binding site. Another aspect is the question whether there is a global preference for the occurrence of discriminatory positions

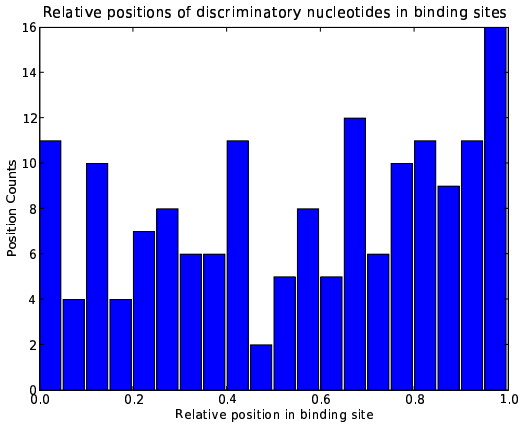


Figure 5.6: *Relative positions along the binding sites of positions which discriminate the two subgroups for 41 TFs.*

along the binding sites. To examine this, we considered the relative positions within the binding site of all positions which are discriminatory. Fig. 5.6 shows the distribution of relative positions for the 41 TFs with CSI structure. As can be seen the discriminatory positions show no clear preference in their positioning along the binding sites.

While these pictures and observations seem to hint at potentially very interesting biological findings, it would require a experimental validation to determine whether these binding motive subgroups are of direct biological relevance. Unfortunately, this kind of validation is beyond the scope of this work.

For a more detailed discussion of all the results refer to section 8.2.

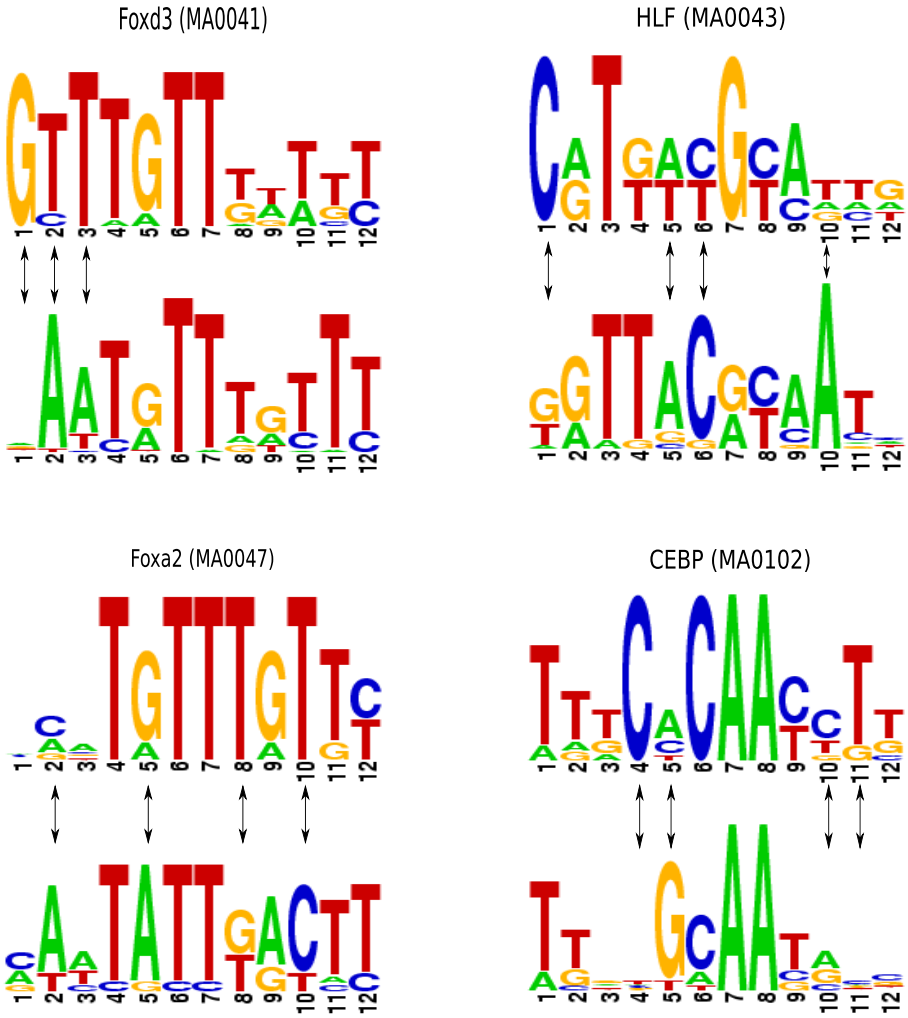


Figure 5.7: *Example sequence logos of binding site subgroups for four TFs from the JASPAR data set.*

Chapter 6

Clustering of Protein Families Using Mixtures

In this chapter we describe the application of CSI mixture based clustering on protein subfamily discovery and simultaneous prediction of functional residues.

6.1 Introduction

Proteins within the same family commonly fall into sub categories which differ by functional specificity. The categorization and analysis of these subgroups is one of the central challenges in the study of these families. In particular it is of interest which residues determine functional specificity of a subgroup. These functional residues are characterized by a strong signal of subgroup specific conservation.

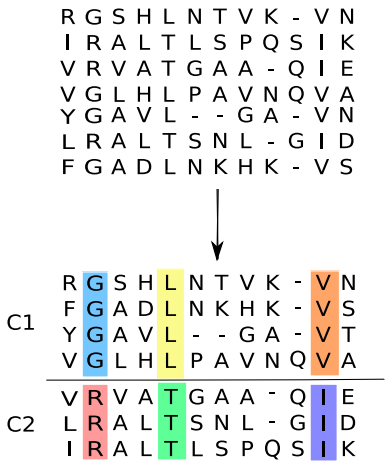


Figure 6.1: Top: *The input to the method is a MSA of protein sequences.* ***Bottom:*** *The output is a clustering into sub-families (C1, C2) and annotation of putative functional residues (colored columns)*

The general problem addressed in this chapter is visualized in Fig.6.1. Given a multiple sequence alignment (MSA) of protein sequences (top), we discover sub-families of sequences

with different functional specificities (C1, C2) and simultaneously predict residue positions which are causal for these functional differences. This is indicated by the differently colored columns (bottom). Generally speaking, there is an increased sub-family-specific sequence conservation at positions which are relevant with respect to the distinct functions of the sub-families. These positions are highly informative for the characterization of the clusters. Conversely, positions which are not relevant for the functional characterization of the sub-families may show very little variability between sub-families. Such positions are very weakly informative for the clustering and subfamily separation. This situation is a natural fit for the sort of fine grained regularities that can be captured in the CSI structure.

A number of studies have focused on the question how to detect residues which determine functional specificity based on prior knowledge of subtype membership. A review of these methods can be found in [90]. Among the approaches taken were relative entropy based scores [78], classification based on similarity to a data base of functional residue templates [28] and contrasting position specific conservation in orthologues and paralogues to predict functional residues [132]. In [200] the authors use known reference protein 3D structures to find conserved discriminatory surface residues. One major limitation of these *supervised* approaches is the requirement of biological expert annotation of the number of subtypes and subtype assignments for each sequence. This limits the usefulness of these methods to cases where prior biological knowledge is abundant. In the absence of such knowledge the inference of the subgroups becomes one central aspect of the prediction of functional residues. In many cases the subgroup structure of a given family is a direct consequence of evolutionary divergence of homologue sequences. As such it is not surprising that methods based on the phylogenetic tree of a family have been extensively and successfully used to study protein family subgroups [105, 110, 142, 195]. However, the performance of these methods does degrade in cases where the evolutionary divergence between subgroups is large. Moreover phylogeny does not account for situations where functional relatedness of proteins arose from a process of convergent evolution. As such there is a need for additional methods for detection and analysis of the subgroups inherent in a set of related sequences. CSI mixture models are well suited to this application for a number of reasons. First, the probabilistic setup provides a good fit for the noisiness to be expected for MSAs, especially from rather divergent sequences. Second, as mentioned above, the data can be expected to contain many weakly or even uninformative features. In such a situation the CSI structure can greatly increase the robustness of the clustering. Third, the combination of the CSI structure and feature ranking schemes provides a structured and principled way to assess the importance of each residue. This allows to make predictions for function positions in the MSA.

One of the challenges of clustering protein families into subgroups based on the sequence is that the discriminating features one attempts to learn are a property of the structure rather than the sequence. As an example, consider three subgroups with perfect conservation of amino acids Leucine, Isoleucine and Tryptophan respectively at one position. A naïve application of a clustering would consider said position to be highly discriminative for all three groups. Of course, this would be misleading due to the great similarity in chemical

properties between Leucine and Isoleucine which makes them, to some extent, synonymous as far as structure is concerned. To adapt the CSI mixture model for this situation we apply a parameter prior in form of a mixture of Dirichlet distributions (see section 6.2). These Dirichlet mixture priors have been successfully used to improve generalization properties of parameter estimates for probabilistic models for small sample sizes [170]. In the CSI framework a suitably chosen prior additionally acts to guide the structure learning towards distributions indicative of structural differences between the subgroups.

6.2 Dirichlet Mixture Priors

As described in section 3.4, the fit of different models to the data is assessed by the model posterior $P(M|D)$ given by

$$P(\hat{\Theta}, M|D) \propto P(D|\hat{\Theta}, M)P(\hat{\Theta}|M)P(M)$$

again $P(D|\hat{\Theta}, M)$ is the Bayesian likelihood based on the data D (Eq. (2.34)), $P(\hat{\Theta}|M)$ is the parameter prior (Eq. (2.35)), $P(M)$ is the prior over the model structure (Eq. (3.5)) and the $\hat{\Theta}$ are the MAP parameter estimates (Eq. (2.4)).

Recall that

$$P(\Theta|M) = P(\pi)\prod_{k=1}^{K}\prod_{j=1}^{p} P(\theta_{kj})$$

.

Assuming all the θ_{kj} are discrete, a possible choice of parameter prior $P(\theta_{kj})$ is a mixture of Dirichlet distributions. A Dirichlet mixture prior (DMP) over a discrete distribution $\theta_{kj} = (\theta_{kj1}, ..., \theta_{kjM})$ is given by

$$P(\theta_{kj}) = \sum_{g=1}^{G} q_g \; D_g(\theta_{kj}|\alpha_g), \tag{6.1}$$

where D_g is the Dirichlet density Eq. (2.38) parameterized by $\alpha_g = (\alpha_{g1}, ..., \alpha_{gM}), \alpha_{gs} > 0$ and q_g are the mixture weights. The DMP has a number of attractive properties for the modeling of protein families. Not only does the DMP retain conjugacy to the discrete distribution which guarantees closed form solutions for the parameter estimates, it also allows for a great degree of flexibility in the induced density over the parameter space. This allows for the integration of amino acid similarities in the structure learning procedure.

The parameter estimators derivation for the DMP case is an straightforward extension of the single Dirichlet prior case [67].

6.3 Prior Parameter Derivation

In order to apply the DMP framework on the problem of regularizing the structure learning for protein families we have to specify the parameterization of $P(\theta_{kj})$. This includes the choice of G, the q_g and the α_g.

We considered three different approaches to arrive at choices for these parameters,

1. choice of parameters based on a PAM series amino acid substitution probability matrix,
2. use of previously published DMP regularizers [170] based on machine learning techniques and
3. heuristic parameter derivation based on basic chemical properties of the amino acids.

The first approach based on PAM matrices [41] proved problematic in that the PAM matrices are stochastic, whereas the parameters of a Dirichlet are only constrained in that they are positive. This means that the information in a PAM matrix (i.e. the parameter values) is on a different scale and cannot be straightforwardly inserted into a DMP. In practice this makes PAM matrices unsuitable for our application.

As for the second approach, the DMPs in [170] were trained to provide suitable regularization to compensate for small sample sizes. While this is certainly related, it is not quite the same as the kind of regularization we require for the CSI structure learning. Clearly a machine learning approach for specifying the prior parameters would be desirable. This however is not straightforward for two reasons: First, it is not clear how the training data for learning a DMP for this application would have to be assembled and secondly the optimization of DMPs is a difficult problem as many local minima exist [170].

Therefore it seemed prudent to consider an additional prior, which was constructed based on the chemical properties of the twenty amino acids. This prior is appealing as it is based on a rather simple heuristic and therefore the values of the parameters lend themselves to straightforward interpretation. The derivation of the third prior will be described in more detail below.

The impact of an amino acid substitution on the fold of a protein depends on the similarity of the chemical properties of the two amino acids. The more dissimilar the amino acids are, the more pronounced the effect on protein structure will be. The relevant chemical properties can be arranged into a hierarchy of more general and specific properties [116]. The nine properties we consider and the assignment of amino acids is summarized in Table 6.1. The amino acid are denoted by the single letter codes (see appendix C) Table entries 'x' and '·' denote presence and absence of a property respectively. Note that the gap symbol '-' is negative for all properties.

Based on this characterization of the amino acids by their basic chemical properties we construct a DMP as follows: To each of the properties in Table 6.1 we assign a component D_g in the DMP. The parameters α_g are chosen such that α_{gs} is larger if amino acid s has the property. This means we construct nine Dirichlet distributions which give high

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	-
Hydrophobic	x	·	·	·	x	·	·	x	x	x	x	x	x	x	·	·	x	x	x	x	·
Polar	·	x	x	x	·	x	x	·	x	·	·	x	·	·	·	x	x	x	x	·	·
Small	x	·	x	x	x	·	·	x	·	·	·	·	·	·	x	x	x	·	·	x	·
Tiny	x	·	·	·	·	·	·	x	·	·	·	·	·	·	·	x	·	·	·	·	·
Aliphatic	·	·	·	·	·	·	·	·	·	x	x	·	·	·	·	·	·	·	·	x	·
Aromatic	·	·	·	·	·	·	·	·	x	·	·	·	·	x	·	·	·	x	x	·	·
Positive	·	x	·	·	·	·	·	·	x	·	·	x	·	·	·	·	·	·	·	·	·
Negative	·	·	·	x	·	·	x	·	·	·	·	·	·	·	·	·	·	·	·	·	·
Charged	·	x	·	x	·	·	x	·	x	·	·	x	·	·	·	·	·	·	·	·	·

Table 6.1: *The twenty amino acids can be characterized by nine chemical properties. A x in the table denotes the presence, a · the absence of a trait.*

density to distributions with strong prevalence of amino acids with a certain property. The combination of all property specific D_g in the DMP then yields a density which allows the quantification of similarity between amino acids in the probabilistic framework. In order to arrive at a scheme to choose the parameters of the DMP the following constraints were taken into consideration:

- The strength of a Dirichlet distribution prior D_g is determined by the sum of its parameters $|\alpha_g|$. The size of $|\alpha_g|$ is also anti-proportional to the variance of D_g. To assign equal strength to all property specific Dirichlets D_g, all $|\alpha_g|$ are set to be identical.
- More general properties should receive greater weights q_g in the DMP.
- The strength of the prior, i.e. $|\alpha_g|$ should depend on the size of the data set N.

This leads to the following heuristics for choosing the DMP parameters: Let the strength of each D_g be one tenth of the data set size; i.e. $|\alpha_g| = \frac{N}{10}$ and $b = \frac{0.75\ |\alpha_g|}{21}$ the base value for the parameters α_g. Then $\alpha_{gs} = b$, for all amino acids were the property is absent and

$$\alpha_{gs} = b + \frac{0.25\ |\alpha_g|}{B_g},$$

for all amino acids where the property is present, where B_g denotes the number amino acid which have the property. Finally, the weights q_g are set to

$$q_g = \frac{B_g}{\sum_{g=1}^{G} B_g}$$

which means that more general properties receive proportionally higher weight in the prior. Thus, the priors in the model introduce two types of bias' into the structure learning. An unspecific preference for a less complex model given by $P(M)$ and a specific preference for parameters $\theta_{X_j|g_{jr}}$ that match the amino acid properties encoded in the prior $P(\theta|M)$.

6.4 Feature Ranking

To predict which features are functional residues for a given subgroup, it is necessary to refine the information in the CSI structure matrix by ranking the informative features. Since these features are distinguished by subgroup specific sequence conservation, the relative entropy is a natural choice to score for putative functional residues. Therefore the score Eq. (3.21) defined in section 3.6.2 was used. Note that the ranking scheme is somewhat similar to the setup used in [78]. The major difference being that in [78] subgroup assignments were assumed to be known and in this work the scoring is based on the posterior distribution of component membership and parameter estimates induced by the expected sufficient statistics in the sEM framework.

6.5 Results

We evaluated the performance of CSI mixture models for protein subfamilies on a number of data set of different sizes from families with known subtype assignments and structural information. This allows for a validation of the clustering results. Any column in the alignment with more than 33% gaps was removed prior to the clustering. Model selection was carried out using the NEC (2.32). The strength of the structure prior was chosen by $\delta = 0.1$ (see section 4.3.1). The position of predicted functional residues in the three dimensional structure can be evaluated from known structures obtained from the *Protein Data Bank* (PDB) [13].

To assess the impact of our DMP on model performance the Accuracy (see section 2.3.2) of the clusterings with DMP were compared to mixtures with the same number of components but a simple uninformative single Dirichlet prior and the UCSC-DMP 'up9' obtained from the supplementary materials of [170].

6.5.1 L-lactate Dehydrogenase Family

This data set consisted of members of the L-lactate dehydrogenase family with differing substrate specificities. The two subfamilies under consideration were the malate and lactate dehydrogenases. In this family, despite substantial variance within and between the subfamilies, a single position is responsible for defining substrate specificity. Taking PDB 1IB6 as reference sequence, an R in position 81 confers specificity for lactate whereas a Q in the same position would switch substrate specificity to malate. Clusterings were computed for the 29 sequences in the PFAM [51] seed alignment of that domain (PF00056). The alignment contained 16 lactate dehydrogenases (LDH) and 13 malate dehydrogenases (MDH).

As shown in Fig. 6.2, the NEC model selection clearly indicated 2 components to provide the best fit for the data.

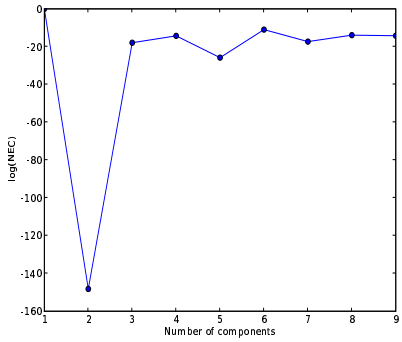


Figure 6.2: *NEC model selection plot for the malate and lactate dehydrogenase data set. The optimal model was given obtained for* $K = 2$.

The two components separated the MDH/LDH groups without error for our DMP mixture.When using the uninformative prior, considerably lower accuracies of around 75% were achieved. To assess the robustness of this result we repeatedly trained two component models with DMP, uninformative and UCSC priors. Averaged over 10 models our DMP achieved accuracy 94% (SD 2.1%). The results of the UCSC-DMP were comparable accuracy-wise although there was no model which provided perfect separation of MDH/LDH. The uninformative prior yielded an accuracy of 76% (SD 8.3).

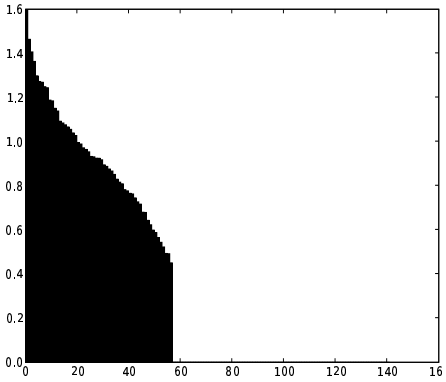


Figure 6.3: *Feature ranking scores for the Malate/Lactate dehydrogenase data set.*

Thus, the CSI mixtures successfully identified the two subfamilies correctly without any prior biological knowledge. The position identified as most informative for distinguishing the groups by the feature ranking shown in Fig. 6.3 was indeed the one responsible for substrate specificity. Many of the other highly ranked residues were arranged around the NAD interaction site of the domain, which suggests they may play a role in malate / lactate recognition.

Fig. 6.4 shows the structure of 1IB6 with the true specificity determining residue (red), other putative functional residues (orange) and the ligand interacting sites (white).

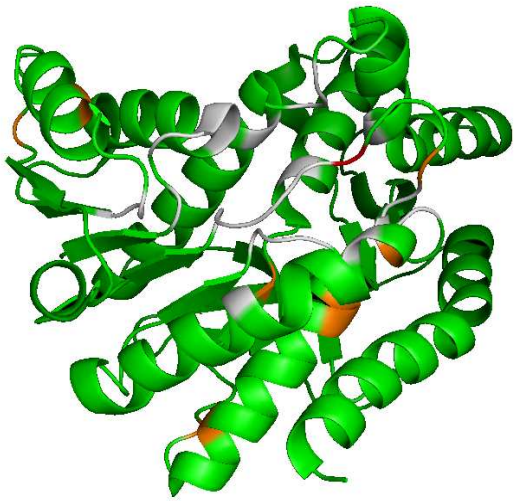


Figure 6.4: *Structure of PDB 1IB6 with predicted functional residues. The true specificity determining site is shown in red, other putative functional residues in orange and additional known ligand interacting sites in white.*

6.5.2 Protein Kinase Family

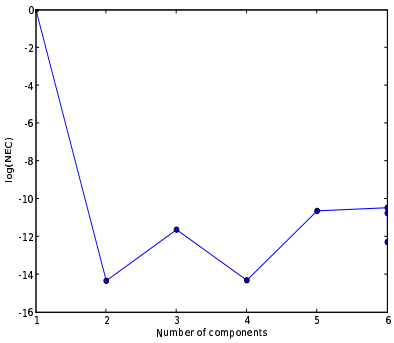


Figure 6.5: *Model selection plot for the kinase data set. The best scores are achieved for* $K = 2$ *and* $K = 4$.

The protein kinase super family is one of the largest and best studied protein families. The human genome contains more than 500 protein kinases [122], many known to be involved with diseases such as cancer or diabetes. The probably most prominent classification of this key players in signal transduction is between tyrosine and serine/threonine kinases. These can be further subdivided according to different regulatory mechanisms [89]. For this data set, these levels of classification were combined by joining tyrosine kinases (TK) with two groups of serine threonine kinases, STE (Homologs of yeast Sterile 7, Sterile 11, Sterile 20 kinases) and AGC (Containing PKA, PKG and PKC families). An alignment of 1221 representative sequences of the subfamilies was obtained from the *Protein kinase*

resource [171].

Fig. 6.5 shows the NEC model selection scores for this data set. It can be seen that the three best NEC model selection scores were assigned to 2,3 and 4 components. Since the scores of 2 and 4 were too similar for a clear choice of components and 3 is the intermediate value, we will consider the results for all three as values of K. For the two component model the TK and STE sequences were collected in one subgroup and the second was almost exclusively AGC. The four component model yielded a clustering in which the sequence of the AGC subfamily got split over two components. In the three component model each family acquired its own subgroup with an accuracy of 87%. Results for the uninformative and UCSC-DMP priors were only slightly worse (about 1% accuracy) for this data set. These results were highly robust in the repetitions with standard deviations of 0.1%-0.3% for the accuracies of all three prior types. In the following PDB 2cpk (cAMP-dependent

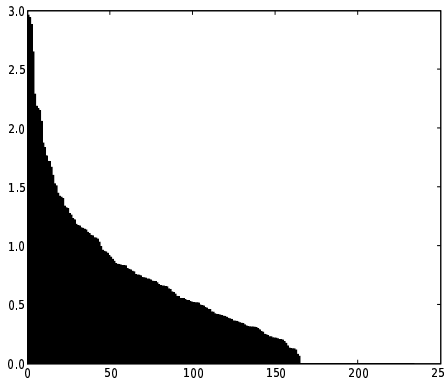


Figure 6.6: *Feature ranking scores for the Kinase cyclase data set.*

protein kinase, alpha-catalytic subunit, Mus musculus) is used as reference sequence for residue numbering. A ranking of the informative features of the three component model with respect to the TK subgroup as shown in Fig. 6.6 yielded within the top 20 positions a region of three residues (168-170) which has been experimentally shown to be important for kinase substrate specificity [77].

Fig. 6.7 shows the structure of PDB 2CPK with the predicted functional residues. The three residue stretch marked in red have been experimentally verified to be of relevance for kinase substrate specificity.

This data set highlights the difficulty of the model selection task. While from the way the data set was constructed three components seemed the right answer, the NEC judged a merging of the TK and STE sequences as well as a further split of the AGC subgroup to capture the data better. Despite that, the three component clustering gave a good separation of the true subgroups and biologically relevant positions in the ranking. This can be seen as a cautionary tale, that while model selection criteria are certainly helpful for selecting K, in cluster analysis of biological data it is often also necessary to look at the different suboptimal models and which regularities they capture.

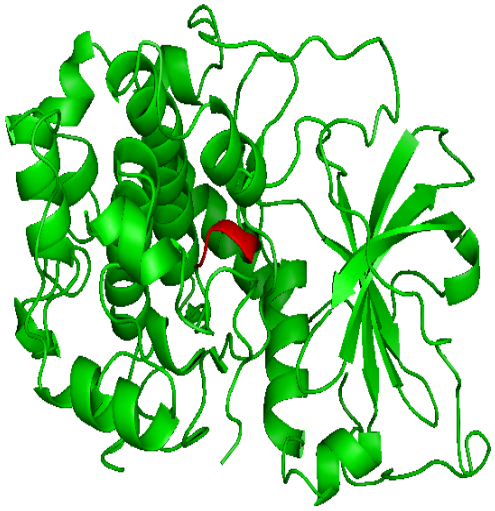


Figure 6.7: *Structure of PDB 2cpk with predicted functional residues. The stretch of three amino acids shown in red has been experimentally shown to be relevant for kinase substrate specificity.*

6.5.3 Nucleotidyl Cyclase Family

Nucleotidyl cyclases play an important role in cellular signaling by producing the secondary messengers cAMP and cGMP which regulate the activity of many other signaling molecules. As cGMP and cAMP fulfill different biological roles, specificity of converting enzymes is imperative.

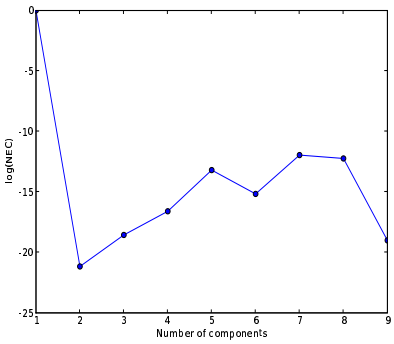


Figure 6.8: *Model selection plot for the cyclase data set. Two components achieve the optimal score.*

Five residues have been experimentally confirmed by substitution experiments to convey substrate specificity. These positions are 938, 1016, 1018, 1019 and 1020 (numbering according to PDB 1AB8) [115]. We used this family as a test case for families with multiple sites involved in functional classification, complementing the L-lactate dehydrogenase

family with a single site.

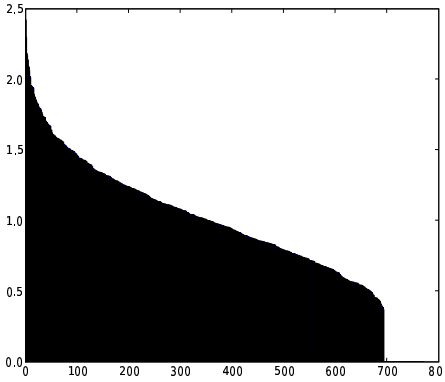


Figure 6.9: *Feature ranking scores for the Nucleotidyl cyclase data set.*

We computed a MSA from 132 GC (EC 4.6.1.2) and AC (EC 4.6.1.1) sequences obtained from the ExPASy data base [60]. As shown in Fig 6.8, the NEC model selection indicated two components to provide the best fit. The model with optimal NEC produced a clustering with an accuracy of 85% with respect to the GC / AC subgroups. Averaged over 10 models the uninformative prior yielded a decreased performance of 63% (SD 6.4) accuracy. For the UCSC prior 'uprior.9comp' the averaged results over 10 repeats were accuracy of 70% (SD 2.5%). For our DMP the averaged results were an accuracy of 73% (SD 4.0). Based on the

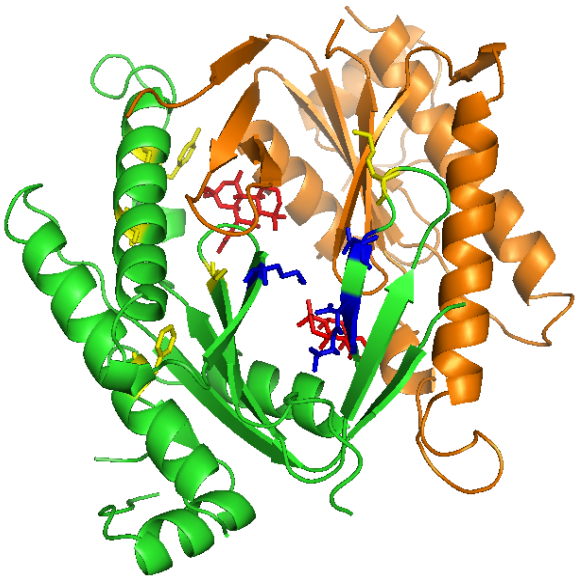


Figure 6.10: *Adenylyl cyclase with predicted functional sites highlighted - Subunit I in green, subunit II in orange. The 10 most informative sites were selected. Shown in red: experimentally validated identified sites, blue: additional identified sites.*

ranking of alignment positions shown in Fig. 6.9, Figure 6.10 shows the three dimensional structure of 1AB8 with the 10 most informative sites highlighted. These contain 4 of the

sites involved in substrate specificity (1018 (ranked 2.), 1016 (3.), 938 (6.), 1019 (9.)). Further top ranking positions included sites which are part of the subunit I and II domain interface (919, 912, 911). Position 943 is right next to a forsoklin interaction site and position 891 interacts with magnesium. Residue 921 finally, is also a metal interacting site [202]. Thus, not only known substrate specific sites were identified, but also further functional sites. The next step would then require experimental validation of the identified sites with no functional annotation.

6.5.4 Partially-supervised Protein Clustering

To investigate the impact of the partially-supervised setup described in section 2.5, we performed clustering on two data sets with different amounts labeled samples. The labeled samples for each class were chosen randomly based on the true biological sequence annotations. In this section the amino acid property prior from section 6.3 was used.

SH3 domain family

The src homology domain 3 (SH3) is a protein interaction module binding to polyproline regions. Its preferred binding partner is characterized by a structural motive, the polyproline type II helix. Two types of binding mode can be distinguished based on the direction of the helix in the binding groove [125]. These different binding preferences are not caused by two different binding patches on the domain, rather it is one site responsible for these interactions. Obviously, this makes an automated classification and identification of specificity inducing sites challenging. We analyzed a large scale interaction study on 20 yeast SH3 domains [189]. Here, each domain was classified into three groups (I, II and Unusual) based solely on their ligands and the labels were chosen from this functional annotation.

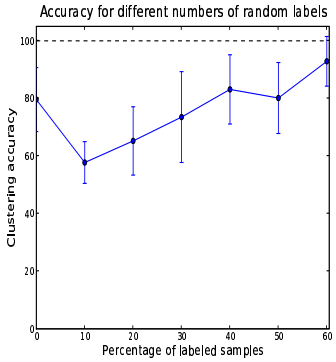


Figure 6.11: *Average accuracy for different numbers of randomly chosen labels for the SH3 domain data set.*

To evaluate the performance for different numbers of labels, average accuracies over thirty repetitions were computed. The results are shown in Fig. 6.11. It can be seen that for 10% labels an average accuracy of around 65% (SD 12%) was observed. When contrasting this

with the unsupervised approach (0 labels) where we observed average accuracies of 80% (SD 11%), it becomes apparent that for $< 30\%$ labels the partially-supervised setup had a detrimental effect on clustering performance on this data set. The main reason for this is probably the random selection of labels for the partially supervised parameter training. It has been observed in a number of studies, that the quality of the labeling is crucial for the performance of partially-supervised approaches [39]. When samples are selected for labeling which are outliers in their own class, the labeling might even have a negative impact on the clustering performance. This issue is increasingly receiving attention in the machine learning community [72].

Pyridoxal-dependent decarboxylase

The pyridoxal-dependent decarboxylase domain is involved as a catalytic coenzyme in a multitude of reactions, including decarboxylation, deamination and transamination primarily in amino acid biosynthesis and metabolism. We considered a data set of pyridoxal-dependent decarboxylase sequences with specificity for either L–glutamate or L–tyrosine substrates. The data set was constructed by selecting all sequences of the PFAM family *pyridoxal_deC* which had annotation for the substrate specificity in the **CATHALYTIC ACTIVITY** field of the corresponding SWISSPROT entries. This resulted in 35 sequences with glutamate specificity and 37 sequences with specificity for tyrosine. An alignment of these 72 sequences was obtained using the *clustalw* [104] software.

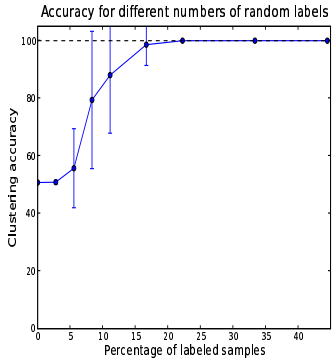


Figure 6.12: *Average accuracy for different numbers of randomly chosen labels for the pyridoxal-dependent decarboxylase data set.*

When clustering without labels the separation of the glutamate and tyrosine subclasses proved to be very challenging. The average performance over 30 repetitions was an accuracy of 51% (SD: 0.3%). When adding the power of the partially supervised framework to the clustering by randomly selecting different numbers of labels for the two subclasses, a different picture emerges. The average accuracies based on 30 repetitions for different amounts of randomly selected labels per class are shown in Fig. 6.12. It can be seen that the clustering accuracy increases monotonously with the number of labeled samples. However, it can also be seen that while the average accuracy improves significantly in the range

of 5%–16% labeled samples, the variance also increases. Again, this is most likely an effect of the random selection of labels. For 22% labels it can be seen that the variance decreases again and models with perfect accuracy and zero variance (over the 30 repetitions) are obtained for $> 22\%$ labels.

The wholly unsupervised clusterings returned a very low-variance, highly robust grouping of the sequences. While these groupings did not reflect the tyrosine/glutamate subgroups with any accuracy, the question was whether they represent some other biological context. Upon examination it became clear that the unsupervised clustering split the data set based on phylogenetic divergence. One cluster contained predominantly achae and bacteria sequences, the other metazoa and viridiplantae (i.e. green plants). Based on this taxonomic classification of the sequences, the clusterings had an average accuracy of 82%. This means that for the unsupervised setup, the clustering picked up on decomposition of the sequences which, while being biologically meaningful in itself, did not reflect the specific question we were interested in. This problem was overcome by including prior knowledge in form of sequence labels.

These results illustrate how the partially supervised approach can improve the parameter estimation and structure learning by guiding it away from local maxima which are not consistent with the biological question under consideration. However, the high variance for moderate amounts of labeled samples again underline the importance of the label selection procedure.

For a discussion of these results refer to section 8.3.

Chapter 7

Clustering of Heart Disease Phenotype Data

In this chapter CSI mixture model based clustering is applied to a data set of heart disease phenotypes.

7.1 Introduction

Defects in heart development during embryogenisis are among the most common birth defects in humans. The genetic basis of the various cardiac anomalies is not yet well understood but there is a rapidly growing number of transcription factors which are implicated in heart development [33]. An increasing number of candidate gene mutations have been identified to be of relevance for heart disease phenotypes [59, 164, 174, 192]. However, a necessary requirement for the success of such a mapping is a clear description of the phenotype to be studied. Clustering can provide such descriptions by identifying subgroups of patients with distinctive phenotype patterns in phenotypically diverse data sets.

In the following we give a brief introduction into the physiology of the human heart. For more details refer to textbooks on cardiovascular medicine (e.g. [24, 54]). The schematic organization of the human heart is shown in Fig. 7.1. The heart consists of four chambers, left atrium (LA), left ventricle (LV), right atrium (RA) and right ventricle (RV). The chambers, as well as the in- and out going blood vessels, are separated by a system of valves. The oxygen poor blood arrives from the body circulation in the RA, passes the tricuspid valve into the RV and from there is pumped through the pulmonary valve (PV) and pulmonary artery to the lungs. The oxygenated blood from the lungs arrives in the LA, passes the mitral valve into the LR and is released through the aortic valve into the aorta. In the healthy heart, the circulatory system is separated into a low–pressure (venous system, RA and RV) and high–pressure (arteric system, LA and LV) system.

Normal heart function can be impaired or even impeded by a variety of anatomical malformations. These include anatomical features such as irregular connections or positions of blood vessels. For instance the aorta may be connected to both ventricles as opposed

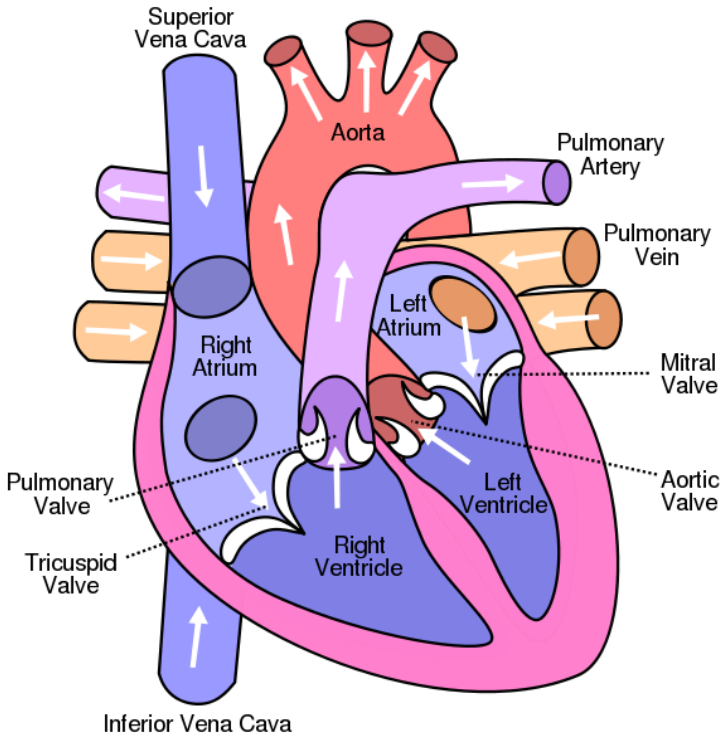


Figure 7.1: *Schematic representation of the human heart. Reproduced from* Wikipedia.org

to only the LV. Another category of possible defect are holes in the membranes separating the chambers (septums). Finally, valves could be narrowed (stenotic), permanently closed (atretic) or leaky. A leaky valve permits blood flow in the closed setting (insufficient). The most common congenital defects in human is a bicupsid aortic valve with two instead of three cusps. Generally speaking, the effect of these abnormalities is an impaired pumping capacity of the heart, leading to reduced supply of oxygen to the body.

When clustering phenotype data from complex disease such as congenital heart defects, the main aim to find distinctive phenotype patterns which characterize subgroups of patients. Due to the high variability in phenotypes often observed in such data sets, it must be expected that there is no clear separation of clusters based on a few features. Rather the patterns characterizing the clusters will arise from the combination of many features. In such a situation the CSI structure matrix can greatly facilitate the practical use of the clustering by making explicit which regularities describe each cluster.

7.2 Data Set

In this section we introduce the data set and describe the phenotypes which are covered.

The specific disease phenotype of each of the 65 individuals in the data set is described by 26 binary features [168, 187] shown in Tab. 7.1. Each of the features gives the presence

or absence of a phenotypic trait with relevance for heart disease indication. Most of the features concern either specific anatomical abnormities or blood pressure indicators. The features are summarized in Tab. 7.1.

7.3 Results

The clustering was performed using a structure prior given by $\delta = 0.05$ (see section 4.3.1). NEC model selection was performed for $K = (1, ..., 10)$. Fig. 7.2 shows the NEC scores for different numbers of components. It can be seen that the four component model obtained the best score on this data set. The cluster sizes were 7, 17, 18 and 23 samples for clusters 1, 2, 3, and 4 respectively.

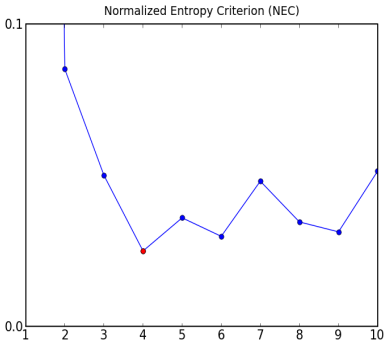


Figure 7.2: *NEC model selection for the heart disease data. It can be seen that four components obtain the optimal score.*

The next question was which features were most informative for the clustering. This was addressed by ranking the 26 features using the entropy criterion Eq. (3.20). Fig. 7.3 shows the ranked features and corresponding scores. Note that the seven features on the bottom were found to be uninformative during the structure learning and obtained a score of 0. Feature **IVS.shunt.BIRIGHTLEFT** was found to be the most strongly discriminative feature.

In order to analyze which feature characterize each cluster we consider the CSI matrix returned by the structure learning. Fig. 7.4 shows a visualization of the CSI matrix of the four clusters. The colors represent the probability for the presence of a feature, the corresponding value in percent is given in each cell of the matrix. The ordering of features is given by the ranking in Fig 7.3. It can be seen for instance, that the most discriminative feature **IVS.shunt.BIRIGHTLEFT**,i.e. the presence or absence of a bidirectional shunt in the interventricular septum, strongly separates the clusters into groups $(1, 2)$ and $(3, 4)$. It is worth noting that this group structure could not be captured by the *default table* CSI formulation (section 3.2). This illustrates how the richer CSI formulation used in our approach is more suited to capture regularities in real biological data.

Feature	Description
AoArch.anatomy.RIGHT.AORTIC.ARCH	Aortic arch (AoArch) is shifted to the right
AoAsc.source.RV	Aorta ascendensis (AoAsc) is sourced in right ventricle (as opposed to the left ventricle in normally developed hearts)
AoAsc.source.RV.AND.LV	AoAsc is sourced in both right and left ventricle
AoV.insufficiency.yes	Aortic valve (AoV) is insufficient
BPsys.r.arm.above	Heightened systolic blood pressure (measured in the right arm)
Concordance.AV	Atrioventricular (AV) concordance, i.e. the right and left atrium is connected to the right and left ventricle respectively
IAS.defect	Interatrial septal defect (IAS)
IAS.defect.PFO.PFO	Patent foramen ovale (PFO), specific defect in the interatrial septum
IAS.defect.secundum.SECUNDUM	Interatrial septal defect type II
IVS.defect.perimembr.DEFECT	Defective perimembrane in interventricular septum (IVS)
IVS.shunt.BIRIGHTLEFT	Bidirectional shunt in the IVS
IVS.shunt.LEFT.RIGHT	Left-to-right shunt in the IVS
LSVC.present.ABNORMALLY.PRESENT	Abnormally formed left superior caval vein
MPA.morphology.DILATED	Dilation of the main pulmonary artery (MPA)
MPA.morphology.STENOTIC	MPA is stenotic
PV.Psys.gradient.increased	Increased systolic blood pressure at the pulmonary valve (PV)
PV.morphology.ATRETIC	PV is atretic
PV.morphology.STENOTIC	PV is stenotic
PV.morphology.bicuspid.BICUSPID	PV is bicuspid
RA.morphology.DILATED	RA is dilated
LV.morphology.DILATED	LV is dilated
LV.morphology.trophy.HYPERTROPHIC	LV is hypertrophic, i.e increased in mass and size
RV.anatomy.INFUNDIBULAR.STENOSIS	RV is stenotic
RV.morphology.DILATED	RV is dilated
RV.morphology.trophy.HYPERTROPHIC	RV is hypertrophic
RV.sys.pressure.increased	Increased systolic blood pressure int the RV

Table 7.1: *The 26 phenotypic features of the heart disease data set.*

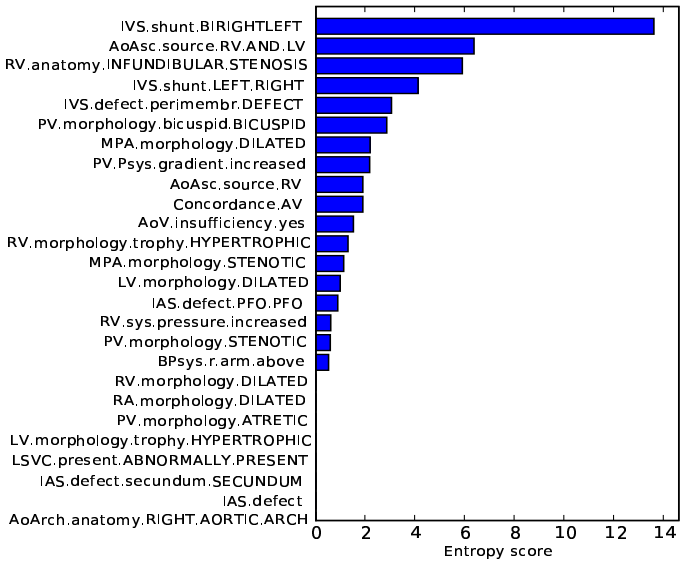


Figure 7.3: *Ranking of the 26 heart disease features. A large score means a feature is strongly discriminative for the clustering.*

There are a number of features whose presence or absence uniquely characterizes a cluster. For cluster 1 these features are

- **PV.Psys.gradient.increased**
- **AoAsc.source.RV**
- **Concordance.AV**.

For all these features there is a stronger disposition towards the presence of the feature than in the other three clusters (e.g. 99% versus 44% for **V.Psys.gradient.increased**). For cluster 2 **BPsys.r.arm.above** shows a characteristic increase of the prevalence of high blood pressure as compared to the other clusters. Cluster 3 is not uniquely characterized by a single feature. Finally, cluster 4 has a unique distribution for features

- **IVS.defect.perimembr.DEFECT**,
- **RV.morphology.trophy.HYPERTROPHIC**
- **RV.sys.pressure.increased**.

In opposition to the other clusters, cluster 4 is characterized by a reduction in prevalence of these three phenotypes. Aside from the features which uniquely characterize a cluster, there are also many features with two groups in the CSI structure. In addition to the previously mentioned **IVS.shunt.BIRIGHTLEFT**, these are

- **AoAsc.source.RV**
- **RV.anatomy.INFUNDIBULAR.STENOSIS**
- **IVS.shunt.LEFT.RIGHT**

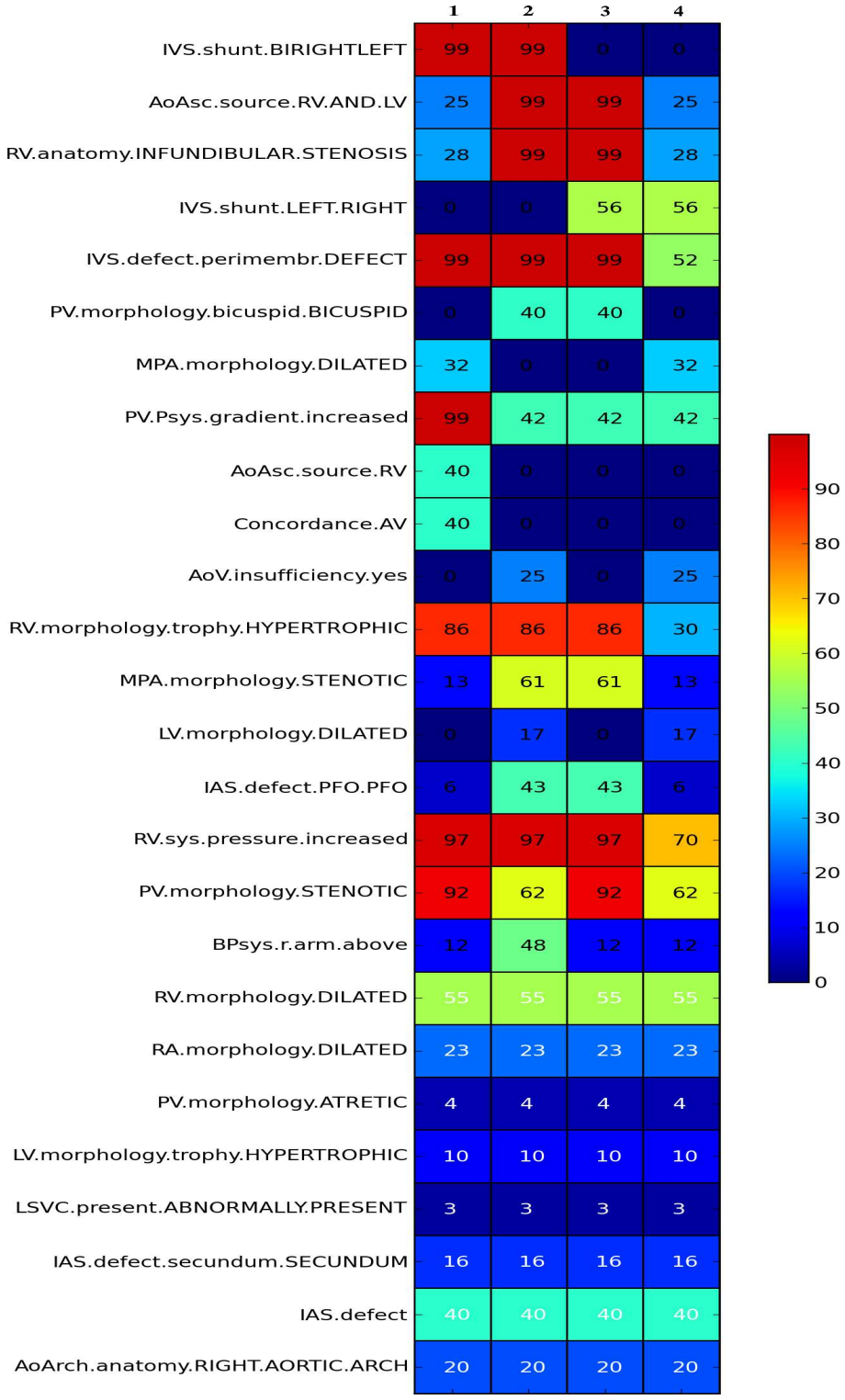


Figure 7.4: *CSI structure matrix of the heart disease data set and four clusters. The colors indicate the probability of observing the presence of a specific phenotype. This probability (in percent) is also given in the cells of the matrix. The order of features is given by the ranking in Fig 7.3.*

- **PV.morphology.bicuspid.BICUSPID**
- **MPA.morphology.DILATED**
- **AoV.insufficiency.yes**
- **MPA.morphology.STENOTIC**
- **LV.morphology.DILATED**
- **IAS.defect.PFO.PFO**
- **PV.morphology.STENOTIC**

From this it is quite clear that, aside from the uniquely discriminatory features, the full characteristics of each cluster arise from a combination of the different feature-specific groups found.

The clustering described above defines four subgroups of heart disease patients with characteristic phenotype patterns. The next question was whether these patterns match with previously described forms of heart disease. It can be seen that cluster 3 matches the phenotype for a classical *tetralogy of Fallot* (TOF) [24] type heart disease. The TOF is characterized by four anatomical abnormalities (therefore *tetra*logy). These abnormalities are a pulmonary stenosis (**PV.morphology.STENOTIC**), a ventricular septal defect (**IVS.defect.perimembr.DEFECT**), an overriding aorta, i.e. the aorta is source in both RV and LV (**AoAsc.source.RV.AND.LV**) and hypertrophy in the RV (**RV.morphology.trophy.HYPERTROPHIC**). While taken singularly these features are also present in other clusters, cluster 3 is the only one which has prevalence for all of the four features and thus matches the clinical definition.

Cluster 1, on the other hand, falls into the clinical definition of the *double outlet right ventricle* (DORV) [24] type heart defect. The DORV is rather broadly defined and can take various forms, in particular the DORV physiology can resemble TOF. Characteristic features of DORV are a bi–directional shunt in the IVS (**IVS.shunt.BIRIGHTLEFT**), an increased blood pressure at the PV (**PV.Psys.gradient.increased**) as well as an increased prevalence of discordant heart anatomy (**Concordance.AV**).

When considering cluster 2, it can be seen that its phenotype pattern is similar to both TOF and DORV type heart disease, without quite matching the criteria for either. This is interesting since, as previously mentioned, the definition of DORV is rather broad and to some degree overlaps with TOF. In fact, in a somewhat simplified view of the clinical situation one can think of TOF and DORV being on the ends of a continuum of heart disease phenotypes. The phenotype patterns of the patients in cluster 2 then marks a point along this continuum. This is an interesting result, since using the classical diagnostic schemes of DORV and TOF, these patients would not fall in either category and might therefore be ignored in subsequent analysis'.

Finally, cluster 4 is largely characterized by the absence of disease phenotypes when compared to the other three clusters. To put it differently, the phenotype pattern is rather diffuse with many features being present with low probability. Therefore the patients in this cluster

can be characterized as a *diverse* group, where no clear phenotypic pattern emerged.

In summary, it can be stated that the phenotype patterns picked up by the clustering both relate closely to established clinical knowledge as well as suggest a new previously undescribed pattern of heart disease phenotypes.

As the original publication of the data set [187] also gave results for a clustering of this data based on a simple hierarchical clustering, we can contrast the two clustering solutions. The agreement of the two clusterings can be quantified by computing the accuracy (section 2.3.2) and treating the labels from [187] as the true classes. This yields an agreement of 78%, i.e. the two clustering disagree on 22% of the pairwise assignments. Also, the chosen cutoff in [187] yielded a more fine grained decomposition with seven clusters and the interesting grouping in cluster 2 was not picked up.

For a discussion of these results refer to section 8.4.

Chapter 8

Discussion

Clustering is a first step in the exploratory analysis of many biological data sets. Due to their high-dimensionality and inherent noisiness, many of these data sets make for challenging clustering problems. In this work we presented an extension to mixture models in form of the CSI formalism which allows for automatic adaption of model complexity and offers attractive properties for facilitating data analysis. The method was applied to three biological data sets from different domains and the analysis' yielded ample biological support for the clusterings obtained.

In the following sections we are going to discuss the various aspects of this work and give indications for possible future research avenues.

8.1 CSI Mixture Models & Structure Learning

As was demonstrated by the three applications presented in this work, the CSI mixture framework is a powerful method for clustering of biological data sets with uninformative features and noisy samples. While the applications described focused on discrete valued data, it bears repetition that the framework is not restricted to discrete data. By adopting appropriate atomic distributions the method can be applied also on continuous data or data sets which contain both discrete and continuous features. The CSI structure which is returned as a direct outcome of the structure learning algorithm does not only have attractive theoretical properties, it also has the very practical advantage of making explicit which features characterize and discriminate the various clusters. This is of considerable importance especially for large data sets with many features. When contrasting our CSI formulation with *default tables*, it can be said that the richer formulation used in this work is more suited to the complexity of real biological data sets (see also sections 7.3 and 8.4).

The combinatorial complexity of the structure learning problem necessitates the use of greedy local search methods for real world data sets. The evaluation of these greedy strategies on randomly generated models with known structure shows that in the majority of cases the greedy top–down search performs as well as the exhaustive enumeration. This result allows some confidence that the structure obtained by the top–down search on real world data sets are also useful.

In the future, it would be interesting to adapt the CSI mixtures by extensions such as *infinite* mixture models [114, 152] or more complex component distributions, for instance conditional Gaussians [37, 177] or the discrete equivalent, mixtures of trees [130]. For the structure learning algorithm, alternative approaches to the structural EM algorithm used in this work could be considered. One possible direction such an alternative could arise from is the newly emerging field of model selection by algebraic statistics [139].

8.2 Transcription Factor Data

In this section the results from the application of CSI mixtures to transcription factor binding site data in chapter 5 will be discussed.

The results of the simulation studies in section 5.3.1 show that the CSI formalism yields more parsimonious and robust representations of the motive for TFs that exhibit a position-wise subgroup structure in their binding pattern. The greater parsimony of the CSI model, as compared to conventional mixtures, was demonstrated for a subgrouping of known Leu3 binding sites. In this example CSI required 30% less parameters than a conventional mixture for an equal performance in separating the high and low binding energy subgroups. The analysis of the conserved fraction of predicted binding sites in human upstream regions in mouse presented in section 5.3.3 showed that a two component CSI model is clearly superior to a conventional two component mixture in the case where a more complex model is warranted by the data. This means that learning the CSI structures led to a more biologically meaningful characterization of the binding patterns of the TFs under consideration. For the TFs where the CSI model increased performance, we can assess that the known binding sites apparently exhibited a biologically relevant subgroup structure. The exact nature of the biological mechanisms underlying these subgroups remains elusive at this point. One possible explanation though would be the existence of different conformations of the TFs which show distinct binding patterns.

While CSI outperformed the conventional mixture, a strong advantage of the CSI (or conventional mixture) model over the single PWM model could not be observed on this data set. This was due to the occurrence of spurious structures for TFs with very few known binding sites and the large number of TFs where the single PWM model seems to be appropriate. This makes sense as one would expect the structure learning to be more vulnerable to outliers in situations where data is extremely sparse. The conclusion we draw from this result is twofold: First, CSI is a practical tool for the search for putative TFBS that fits in seamlessly within the probabilistic framework for scoring hits that has been established for the single PWM model (e.g. [107]). For a practical analysis using CSI though it seems important to require a minimum number of available binding sites (say 18) in order to attempt to fit a CSI model and to use the single PWM model otherwise. This could be easily included into the model prior. Secondly, we would expect the general usefulness of the CSI approach to increase in the future as the pool of known confirmed binding sites increases.

For future work, it would be interesting to repeat the clustering of binding sites on larger data sets such as the TRANSFAC data base [124, 198] to obtain a larger sample of TFs which shows subgrouping in their binding sites. This would allow to address more questions about the distribution of CSI structures for transcription factors of the kind discussed in section 5.3.4. This might yield additional insights into the biological mechanism behind these subgroups. Complementary to such high-throughput analysis, it would also be very interesting to examine the subgroups found for specific factors in detail. While this would require dedicated wet lab experiments for most factors, there is also an increasing number of studies which provide detailed information about the binding behavior of specific factors (e.g. [25, 26, 95, 111]). The availability of such data might help to interpret the results of the binding site clusterings.

8.3 Protein Family Data

The results of CSI mixture-based clustering of protein families presented in chapter 6 show that the approach is capable of simultaneously finding biological relevant subgroups, as well as predicting functional residues which characterize these groups. The functional residue prediction proved to be robust to some degree to imperfections in the clustering with respect to the true biological subgroup memberships. This highlights that this kind of analysis is also useful when functional classifications for the proteins of a data set are not, or only partly, available.

The results also show that the use of DMPs in the analysis, consistently lead to an increase in performance on the protein data when contrasted with the uninformative prior. For the two DMP under consideration, the heuristic DMP based on nine chemical amino acid properties performed somewhat better than a previously published DMP on the data sets under consideration, although the difference was rather slight. In addition, due to its rather simple structure, the heuristic DMP allows for an interpretation of the DMP components and by that also the kind of bias it introduces into the parameter estimates.

One observation which could be made was that the degree of improvement of the DMPs over the uninformative prior differs considerably between the families. This is not surprising as one would expect differing amounts of synonymous substitutions within the various subgroups and that is the situation where a DMP makes the largest difference as compared to the uninformative prior. Also, there was no case where the use of a DMP had a detrimental effect on performance, which highlights the general usefulness of the DMP for this kind of data.

For future work it might be worth investigating the impact of different DMPs on the clustering results and in particular whether customized DMPs for specific applications yield improvement over the general purpose priors used in this work. Moreover, now that the usefulness of the method has been established on families with abundant prior knowledge about subgroups and structure, a next step would be be to bring the method to bear to pre-

dict groups and functional residues on data sets where such knowledge does not exist yet. This, again would require experimental validation of the predicted residues and subgroups. Another avenue for validation of predicted functional residues might come from the many alanine substitution studies (e.g. [7, 123, 193, 201]) which experimentally investigate the impact of specific alanine substitutions on the structure and function of proteins.

The preliminary analysis of the partially-supervised setup in section 6.5.4 show that the approach has the potential to guide the clustering to provide groups which are meaningful for specific biological questions. At the same time, the results are also a cautionary tale in that small amounts of labels can actually have a detrimental effect on the clustering, if randomly selected. This is probably the case, due to the random label selection. This can cause the cluster centers to be strongly characterized by outliers, especially if only few labels are available. The effect of low quality labels to have a detrimental effect on clustering performance has been reported in the literature [39]. Bearing that in mind, if the labeled samples are known to be of high quality, the partially-supervised approach can greatly improve the clustering setup.

8.4 Heart Disease Phenotype Data

The clustering of heart disease phenotypes presented in chapter 7 is a good example of how clustering can be used as a first step in the attempt to unravel complex phenotypes. Also, it shows how the CSI matrix (Fig. 7.4) can facilitate cluster analysis by making explicit which regularities characterize each cluster.

The clustering revealed distinctive phenotype patterns with strong correlation to classical heart diseases. In addition to two clusters with direct relation to known clinical phenotypes (TOF and DORV), the clustering also gave a novel intermediate phenotype pattern which does not fit into the classical schemes and might therefore not be taken into account when studying causal genetic variants. Finally, samples which did not show a strong common phenotype pattern were collected in a single cluster and thereby were prevented from confounding the regularities present in the other clusters.

The difficulty of studying the genetic roots of complex phenotypes such as congenital heart disease, is that there are many distinct genetic factors which contribute to a given phenotype and that the different phenotypes overlap. This makes the direct application of genetic association approaches problematic. In this situation the decomposition of samples into distinct phenotype patterns given by a clustering can sharpen the hypothesis' to be tested and thereby increase the power of the analysis. The underlying assumption being, that patients which share distinctive phenotype patterns are more likely to also share underlying causal genetic variants. In this manner clustering can pave the way for subsequent analysis' and deeper understanding of complex phenotypes. Moreover, it should also be noted, that for most complex genetic diseases the established diagnostic categories are by necessity tailored toward clinical treatment. This means these categories do not necessarily reflect

commonality in genetic causes. Due to the unsupervised nature of clustering, it is possible to obtain subgroups in a manner which is not biased by conventional wisdom. One example of that is the potentially interesting compound phenotype found in cluster 2.

Another aspect worth pointing out is that the CSI structure for the heart disease data could not be represented as a *default table* (see section 3.1). This highlights that our CSI formulation and the greater flexibility it affords to the structure search, is a more natural representation of the kind of regularities which can be expected in biological data sets.

When comparing the CSI mixture–based clustering to the hierarchical clustering presented in a previous publication [187], two points seem worth pointing out. While the two clusterings were fairly consistent in the pairwise assignment of the data points, both the number of clusters as well as the interpretation of the clusters differed. The hierarchical clustering returned a more fine grained decomposition into seven clusters for the chosen cutoff. For instance, this clustering split the TOF patients into four subgroups. While such a fine grained separation also can be of interest for other applications, it can be argued that the more general groups returned by the CSI clustering are more promising as an initial step of genetic association studies. This is due to the loss of power in the association analysis that comes with the small sample sizes for too specific clusters.

For future work, it would be interesting to follow up the cluster analysis by performing genetic association [73] and gene expression studies [109] which might establish connections between specific phenotypes and genomic regions, genes or expression patterns. Also, the analysis presented here was based on a subset of 26 phenotypes selected in [187] out of a larger data set of 250 features. Repeating the analysis on the whole data set might yield a clearer characterization of the phenotypes pattern and could be contrasted with the results obtained so far.

Bibliography

[1] M. Abramowitz and I. A. E. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, New York, 1972.

[2] M. Aigner. A characterization of the Bell numbers. *Discrete Math.*, 205:207–210, 1999.

[3] H. Akaike. Information theory and the extension of the maximum likelihood principle. *Second International Symposium on Information Theory*, pages 267–281, 1973.

[4] P. Allison. Multiple imputation for missing data: a cautionary tale. *Sociological Methods and Research*, 28(3):301–309, 2000.

[5] E. B. Andersen. Sufficiency and exponential families for discrete sample spaces. *Journal of the American Statistical Association*, 65(331):1248–1255, 1970.

[6] R. Ariew. *Ockham's Razor: A Historical and Philosophical Analysis of Ockham's Principle of Parsimony,*. Champaign-Urbana, University of Illinois, 1976.

[7] P. Auguste, O. Robledo, C. Olivier, J. Froger, V. Praloran, A. Pouplard-Barthelaix, and H. Gascan. Alanine substitution for Thr268 and Asp269 of soluble ciliary neurotrophic factor (CNTF) receptor alpha component defines a specific antagonist for the CNTF response. *J. Biol. Chem.*, 271:26049–26056, Oct 1996.

[8] J. D. Banfield and A. E. Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, 49(3):803–821, 1993.

[9] Y. Barash, G. Elidan, N. Friedman, and T. Kaplan. Modeling dependencies in protein - dna binding sites. In *Proceedings of RECOMB '03*, pages 28–37, New York, NY, USA, 2003. ACM Press.

[10] Y. Barash and N. Friedman. Context-specific bayesian clustering for gene expression data. *J Comput Biol*, 9(2):169–91, 2002.

[11] K. J.-A. Barger. Mixtures of Exponential Distributions to Describe the Distribution of Poisson Means in Estimating the Number of Unobserved Classes. Master's thesis, Cornell University, Ithaca, New York, U.S, May 2006.

[12] A. Ben-Hur, A. Elisseeff, and I. Guyon. A stability based method for discovering structure in clustered data. *Pac Symp Biocomput*, pages 6–17, 2002.

[13] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, and P. Bourne. The Protein Data Bank. *Nucleic Acids Res.*, 28:235–242, Jan 2000.

[14] C. Biernacki, G. Celeux, and G. Govaert. An improvement of the nec criterion for assessing the number of clusters in a mixture model. *Non-Linear Anal.*, 20(3):267–272, 1999.

[15] C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):719–725, 2000.

[16] C. Biernacki and G. Govaert. Using the classification likelihood to choose the number of clusters. *Computing Science and Statistics*, 29(2):451–457, 1997.

[17] J. Bilmes. A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models, 1997.

[18] Y. Bilu and N. Barkai. The design of transcription-factor binding sites is affected by combinatorial regulation. *Genome Biol*, 6(12):R103, 2005.

[19] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA, 1995.

[20] H. Bolouri and E. H. Davidson. Modeling DNA sequence-based cis-regulatory gene networks. *Dev Biol*, 246(1):2–13, Jun 2002.

[21] C. Boutilier, R. Dearden, and M. Goldszmidt. Exploiting structure in policy construction. In C. Mellish, editor, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1104–1111, San Francisco, 1995. Morgan Kaufmann.

[22] C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. Context-specific independence in Bayesian networks. In *Uncertainty in Artificial Intelligence*, pages 115–123, 1996.

[23] J. Bowie, R. Lüthy, and D. Eisenberg. A method to identify protein sequences that fold into a known three-dimensional structure. *Science*, 253:164–170, Jul 1991.

[24] E. Braunwald, D. Zipes, P. Libby, and R. Bonow. *Braunwald's Heart Disease: A Textbook of Cardiovascular Medicine, Seventh ed.* Saunders, London, 2004.

[25] M. Bulyk, E. Gentalen, D. Lockhart, and G. Church. Quantifying DNA-protein interactions by double-stranded DNA arrays. *Nat. Biotechnol.*, 17:573–577, Jun 1999.

[26] M. L. Bulyk, X. Huang, Y. Choo, and G. M. Church. Exploring the dna-binding specificities of zinc fingers with dna microarrays. *Proc Natl Acad Sci U S A*, 98(13):7158–7163, June 2001.

[27] L. L. Cavalli-Sforza and A. W. F. Edwards. Phylogenetic analysis: Models and estimation procedures. *Evolution*, 21(3):550–570, 1967.

[28] S. Chakrabarti and C. J. Lanczycki. Analysis and prediction of functionally important sites in proteins. *Protein Sci*, 16(1):4–13, Jan 2007.

[29] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[30] S. Chib. Marginal likelihood from the gibbs output. *Journal of the American Statistical Association*, 90(432):1313–1321, 1995.

[31] D. M. Chickering and D. Heckerman. Efficient approximations for the marginal likelihood of bayesian networks with hidden variables. *Machine Learning*, 29(2-3):181–212, 1997.

[32] D. M. Chickering and D. Heckerman. Efficient approximations for the marginal likelihood of bayesian networks with hidden variables. *Mach. Learn.*, 29(2-3):181–212, 1997.

[33] K. L. Clark, K. E. Yutzey, and D. W. Benson. Transcription factors and congenital heart defects. *Annu Rev Physiol*, 68:97–121, 2006.

[34] L. Coin, A. Bateman, and R. Durbin. Enhanced protein domain discovery using taxonomy. *BMC Bioinformatics*, 5:56, May 2004.

[35] R. Copley, T. Doerks, I. Letunic, and P. Bork. Protein domain analysis in the era of complete genomes. *FEBS Lett.*, 513:129–134, Feb 2002.

[36] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, September 1995.

[37] I. Costa, S. Roepcke, and A. Schliep. Gene expression trees in lymphoid development. *BMC Immunol.*, 8:25, 2007.

[38] I. G. Costa, R. Krause, L. Optiz, and A. Schliep. Semi-supervised learning for the identification of syn-expressed genes from fused microarray and in situ image data. *BMC Bioinformatics*, 2007. Accepted.

[39] I. G. Costa and A. Schliep. On the feasibility of heterogeneous analysis of large scale biological data. In *Proceedings of ECML/PKDD 2006 Workshop on Data and Text Mining for Integrative Biology*, pages 55–60, 2006.

[40] N. M. A. David J. Hand. Defining attributes for scorecard construction in credit scoring. *Journal of Applied Statistics*, 27(5):527–540, July 2000.

[41] M. O. Dayhoff. *Atlas of protein sequence and structure*, volume 5. National biomedical research foundation, 1972.

[42] F. De Dombal. The diagnosis of acute abdominal pain with computer assistance: worldwide perspective. *Ann Chir*, 45:273–277, 1991.

[43] N. Dean and A. Raftery. Normal uniform mixture differential gene expression detection for cdna microarrays. *BMC Bioinformatics*, 6(1):173, 2005.

[44] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, pages 1–38, 1977.

[45] A. A. Devalkeneer, P. A. Robe, J. G. Verly, and C. L. M. Phillips. Generalized expectation-maximization segmentation of brain mr images. *Medical Imaging 2006: Image Processing*, 6144(1):61443I, 2006.

[46] A. Dobra, C. Hans, B. Jones, J. R. Nevins, G. Yao, and M. West. Sparse graphical models for exploring gene expression data. *J. Multivar. Anal.*, 90(1):196–212, 2004.

[47] P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning 29*, pages 103–?130, 1997.

[48] M. D. Escobar. Estimating normal means with a dirichlet process prior. *Journal of the American Statistical Association*, 89(425):268–277, 1994.

[49] M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

[50] Y. Feng and G. Hamerly. Pg-means: learning the number of clusters in data. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *NIPS*, pages 393–400. MIT Press, 2006.

[51] R. D. Finn, J. Tate, J. Mistry, P. C. Coggill, S. J. Sammut, H. R. Hotz, G. Ceric, K. Forslund, S. R. Eddy, E. L. Sonnhammer, and A. Bateman. The Pfam protein families database. *Nucleic Acids Res.*, 36:D281–288, Jan 2008.

[52] A. Fossati, E. Arnaud, R. P. Horaud, and P. Fua. Tracking articulated bodies using generalized expectation maximization. In *Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment (CVPR workshop, NORDIA'08)*, june 2008.

[53] C. Fraley and A. Raftery. Bayesian regularization for normal mixture estimation and model-based clustering. Technical Report TR-2005-486, Department of Statistics, University of Washington,Seattle, Box 354322, 2005.

[54] R. Freedom. *Congenital Heart Disease*. Futura Pub. Co, Mount Kisco, 1997.

[55] N. Friedman. Learning belief networks in the presence of missing values and hidden variables. In *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, pages 125–133, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[56] N. Friedman. The bayesian structural em algorithm. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 129–138, 1998.

[57] N. Friedman and M. Goldszmidt. Learning bayesian networks with local structure. In *Proceedings of the NATO Advanced Study Institute on Learning in graphical models*, pages 421–459, Norwell, MA, USA, 1998. Kluwer Academic Publishers.

[58] A. Gammerman and A. R. Thatcher. Bayesian diagnostic probabilities without assuming independence of symptoms. *Methods Inf Med*, 30(1):15–22, 1991.

[59] V. Garg, I. S. Kathiriya, R. Barnes, M. K. Schluterman, I. N. King, C. A. Butler, C. R. Rothrock, R. S. Eapen, K. Hirayama-Yamada, K. Joo, R. Matsuoka, J. C. Cohen, and D. Srivastava. Gata4 mutations cause human congenital heart defects and reveal an interaction with tbx5. *Nature*, 424(6947):443–7, 2003.

[60] E. Gasteiger, A. Gattiker, C. Hoogland, I. Ivanyi, . D. Appel, and A. Bairoch. ExPASy: The proteomics server for in-depth protein knowledge and analysis. *Nucleic Acids Res*, 31(13):3784–8, 2003.

[61] G.Celeux and G.Soromenho. An Entropy Criterion for Assessing the Number of Clusters in a Mixture Model. *J. Classification*, 13:195–212, 1996.

[62] D. Geiger and D. Heckerman. Advances in probabilistic reasoning. In *Proceedings of the seventh conference (1991) on Uncertainty in artificial intelligence*, pages 118–126, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc.

[63] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis, Second Edition*. Chapman & Hall/CRC, July 2003.

[64] B. Georgi. Mixture modeling and group inference in fused genotype and phenotype data. Master's thesis, Free University of Berlin, Berlin, Germany, 2005.

[65] B. Georgi and A. Schliep. Context-specific independence mixture modeling for positional weight matrices. *Bioinformatics*, 22(14):166–73, Jul 2006.

[66] B. Georgi and A. Schliep. Partially-supervised context-specific independence mixture modeling. In *workshop on Data Mining in Functional Genomics and Proteomics, ECML 2007*, 2007.

[67] B. Georgi, J. Schultz, and A. Schliep. Context-specific independence mixture modelling for protein families. In *Knowledge Discovery in Databases: PKDD 2007*, volume Volume 4702/2007, pages 79–90. Springer Berlin / Heidelberg, 2007.

[68] B. Georgi, M. Spence, P. Flodman, and A. Schliep. Mixture model based group inference in fused genotype and phenotype data. In *Studies in Classification, Data Analysis, and Knowledge Organization*. Springer, 2007.

[69] D. Geschwind, J. Sowinski, C. Lord, P. Iversen, J. Shestack, P. Jones, L. Ducat, and S. Spence. The autism genetic resource exchange: a resource for the study of autism and related neuropsychiatric conditions. *Am. J. Hum. Genet.*, 69:463–466, Aug 2001.

[70] S. Glesner and D. Koller. Constructing flexible dynamic belief networks from first-order probalistic knowledge bases. In *Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 217–226, 1995.

[71] P. J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[72] D. Greene and P. Cunningham. Constraint selection by committee: An ensemble approach to identifying informative constraints for semi-supervised clustering. In *Machine Learning: ECML 2007*, volume Volume 4701/2007, pages 140–151. Springer Berlin / Heidelberg, 2007.

[73] A. J. F. Griffiths, J. H. Miller, D. T. Suzuki, R. C. Lewontin, and W. M. Gelbart. *An Introduction to Genetic Analysis*. W. H. Freeman and Company, seventh edition, 2000.

[74] R. Gross, J. Yang, and A. Waibel. Growing gaussian mixture models for pose invariant face recognition. *icpr*, 01:5088, 2000.

[75] S. Haesler, C. Rochefort, B. Georgi, P. Licznerski, P. Osten, and C. Scharff. Incomplete and Inaccurate Vocal Imitation after Knockdown of FoxP2 in Songbird Basal Ganglia Nucleus Area X. *PLoS Biol*, 5:e321, Dec 2007.

[76] D. Hand and K. Yu. Idiot's bayes - not so stupid after all? *International Statistical Review,Vol 69*, pages 385–399, 2001.

[77] S. K. Hanks, A. M. Quinn, and T. Hunter. The protein kinase family: conserved features and deduced phylogeny of the catalytic domains. *Science*, 241(4861):42–52, Jul 1988. Comparative Study.

[78] S. Hannenhalli and R. B. Russell. Analysis and prediction of functional sub-types from protein sequence alignments. *J Mol Biol*, 303(1):61–76, Oct 2000.

[79] S. Hannenhalli and L.-S. Wang. Enhanced position weight matrices using mixture models. *Bioinformatics*, 21 Suppl 1:i204–i212, Jun 2005.

[80] R. Hardison. Comparative genomics. *PLoS Biol.*, 1:E58, Nov 2003.

[81] D. Heckerman. A tutorial on learning with bayesian networks. Technical report, Microsoft Research, Redmond, Washington, 1995.

[82] D. E. Heckerman. *Probabilistic similarity networks*. PhD thesis, Stanford University, Stanford, CA, USA, 1990. Adviser-Edward H. Shortliffe.

[83] K. Hede. Superhighway or blind alley? The cancer genome atlas releases first results. *J. Natl. Cancer Inst.*, 100:1566–1569, Nov 2008.

[84] C. Hennig and P. Coretto. The noise component in model-based cluster analysis. In *Data Analysis, Machine Learning and Applications*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 127–138. Springer Berlin Heidelberg, 2008.

[85] G. Z. Hertz and G. D. Stormo. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15(7-8):563–577, Jul 1999.

[86] G. Z. Hertz and G. D. Stormo. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15(7-8):563–577, Jul 1999.

[87] A. Hinrichs, D. Karolchik, R. Baertsch, G. P. Barber, G. Bejerano, H. Clawson, M. Diekhans, T. S. Furey, R. A. Harte, F. Hsu, J. Hillman-Jackson, R. M. Kuhn, J. S. Pedersen, A. Pohl, B. J. Raney, K. R. Rosenbloom, A. Siepel, K. E. Smith, C. W. Sugnet, A. Sultan-Qurraie, D. J. Thomas, H. Trumbower, R. J. Weber, M. Weirauch, A. S. Zweig, D. Haussler, and W. J. Kent. The UCSC Genome Browser Database: update 2006. *Nucleic Acids Res*, 34(Database issue):590–598, Jan 2006.

[88] J. V. Hulse and T. M. Khoshgoftaar. A comprehensive empirical evaluation of missing value imputation in noisy software measurement data. *J. Syst. Softw.*, 81(5):691–708, 2008.

[89] T. Hunter. Protein kinase classification. *Methods Enzymol*, 200:3–37, 1991.

[90] S. Jones and J. M. Thornton. Searching for functional sites in protein structures. *Curr Opin Chem Biol*, 8(1):3–7, Feb 2004.

[91] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[92] I. Korf. Gene finding in novel genomes. *BMC Bioinformatics*, 5:59, May 2004.

[93] E. A. Kotelnikova, V. J. Makeev, and M. S. Gelfand. Evolution of transcription factor DNA binding sites. *Gene*, 347(2):255–263, Mar 2005.

[94] A. Krogh. Two methods for improving performance of a hmm and their application for gene finding. In *Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology*, pages 179–186. AAAI Press, 1997.

[95] A. Krylov, O. Zasedateleva, D. Prokopenko, J. Rouviere-Yaniv, and A. Mirzabekov. Massive parallel analysis of the binding specificity of histone-like protein HU to single- and double-stranded DNA with generic oligodeoxyribonucleotide microchips. *Nucleic Acids Res.*, 29:2654–2660, Jun 2001.

[96] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[97] D. Kulp, D. Haussler, M. G. Reese, and F. H. Eeckman. A generalized hidden markov model for the recognition of human genes in DNA. *ISMB-96*, pages 134–141, 1996.

[98] L. I. Kuncheva. On the optimality of naive bayes with dependent binary features. *Pattern Recogn. Lett.*, 27(7):830–837, 2006.

[99] G. N. Lance and W. T. Williams. A General Theory of Classificatory Sorting Strategies: 1. Hierarchical Systems. *The Computer Journal*, 9(4):373–380, 1967.

[100] E. Lander and L. Kruglyak. Genetic dissection of complex traits: guidelines for interpreting and reporting linkage results. *Nat. Genet.*, 11:241–247, Nov 1995.

[101] S. Lang. *Calculus of Several Variables*. MA: Addison-Wesley, 1973.

[102] T. Lange, M. H. C. Law, A. K. Jain, and J. M. Buhmann. Learning with constrained and unlabelled data. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 731–738, Washington, DC, USA, 2005. IEEE Computer Society.

[103] T. Lange, V. Roth, M. L. Braun, and J. M. Buhmann. Stability-based validation of clustering solutions. *Neural Comput.*, 16(6):1299–1323, 2004.

[104] M. A. Larkin, G. Blackshields, N. P. Brown, R. Chenna, P. A. McGettigan, H. McWilliam, F. Valentin, I. M. Wallace, A. Wilm, R. Lopez, J. D. Thompson, T. J. Gibson, and D. G. Higgins. Clustal W and Clustal X version 2.0. *Bioinformatics*, 23:2947–2948, Nov 2007.

[105] B. Lazareva-Ulitsky, K. Diemer, and P. D. Thomas. On the quality of tree-based protein classification. *Bioinformatics*, 21(9):1876–1890, May 2005. Comparative Study.

[106] P. F. Lazarsfeld and N. W. Henry, editors. *Latent Structure Analysis*. Houghton Mifflin, Boston, 1968.

[107] S. Levy and S. Hannenhalli. Identification of transcription factor binding sites in the human genome sequence. *Mamm Genome*, 13(9):510–514, Sep 2002.

[108] J. Li and H. Zha. Two-way poisson mixture models for simultaneous document classification and word clustering. *Computational Statistics & Data Analysis*, 50(1):163–180, January 2006.

[109] S. Liang, S. Liang, and R. Somogyi. Tutorial: Gene expression data analysis and modeling. In *Pacific Symposium on Biocomputing 2̌01999 (PSB2̌01999*, 1999.

[110] O. Lichtarge, H. R. Bourne, and F. E. Cohen. An evolutionary trace method defines binding surfaces common to protein families. *J Mol Biol*, 257(2):342–358, Mar 1996.

[111] J. Linnell, R. Mott, S. Field, D. P. Kwiatkowski, J. Ragoussis, and I. A. Udalova. Quantitative high-throughput analysis of transcription factor binding specificities. *Nucleic Acids Res*, 32(4), 2004.

[112] R. J. A. Little and D. B. Rubin. *Statistical analysis with missing data*. John Wiley & Sons, Inc., New York, NY, USA, 1986.

[113] X. Liu and N. D. Clarke. Rationalization of gene regulation by a eukaryotic transcription factor: calculation of regulatory region occupancy from predicted binding affinities. *J Mol Biol*, 323(1):1–8, Oct 2002.

[114] X. Liu, S. Sivaganesan, K. Yeung, J. Guo, R. Bumgarner, and M. Medvedovic. Context-specific infinite mixtures for clustering gene expression profiles across diverse microarray dataset. *Bioinformatics*, 22:1737–1744, Jul 2006.

[115] Y. Liu, A. E. Ruoho, V. D. Rao, and J. H. Hurley. Catalytic mechanism of the adenylyl and guanylyl cyclases: modeling and mutational analysis. *Proc Natl Acad Sci U S A*, 94(25):13414–13419, Dec 1997.

[116] C. D. Livingstone and G. J. Barton. Protein sequence alignments: a strategy for the hierarchical analysis of residue conservation. *Comput Appl Biosci*, 9(6):745–756, Dec 1993.

[117] D. Lowd and P. Domingos. Naive bayes models for probability estimation. In *Proceedings of the International Conference on Machine Learning*, 2005.

[118] M. Z. Ludwig, N. H. Patel, and M. Kreitman. Functional analysis of eve stripe 2 enhancer evolution in Drosophila: rules governing conservation and change. *Development*, 125(5):949–958, Mar 1998.

[119] J. B. Macqueen. Some methods of classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[120] T. K. Man and G. D. Stormo. Non-independence of Mnt repressor-operator interaction determined by a new quantitative multiple fluorescence relative affinity (QuMFRA) assay. *Nucleic Acids Res*, 29(12):2471–2478, Jun 2001.

[121] T. P. Mann. Numerically stable hidden markov model implementation, 2006.

[122] G. Manning, D. B. Whyte, R. Martinez, T. Hunter, and S. Sudarsanam. The protein kinase complement of the human genome. *Science*, 298(5600):1912–1934, Dec 2002.

[123] H. Matthews, N. Audsley, and R. Weaver. Alanine substitution and deletion analogues of manduca sexta allatostatin: Structure-activity relationship on the spontaneous contractions of the foregut of larval lacanobia oleracea. *Journal of Insect Physiology*, 52(2):128 – 135, 2006.

[124] V. Matys, O. Kel-Margoulis, E. Fricke, I. Liebich, S. Land, A. Barre-Dirrie, I. Reuter, D. Chekmenev, M. Krull, K. Hornischer, N. Voss, P. Stegmaier, B. Lewicki-Potapov, H. Saxel, A. Kel, and E. Wingender. TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes. *Nucleic Acids Res.*, 34:D108–110, Jan 2006.

[125] B. Mayer. SH3 domains: complexity in moderation. *J. Cell. Sci.*, 114:1253–1263, Apr 2001.

[126] J. D. McAuliffe, D. M. Blei, and M. I. Jordan. Nonparametric empirical bayes for the dirichlet process mixture model. *Statistics and Computing*, 16(1):5–14, January 2006.

[127] G. McLachlan, R. Bean, and D. Peel. A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics*, 18:413–422, Mar 2002.

[128] G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley, New York, 1997.

[129] G. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley & Sons, 2000.

[130] M. Meila and M. I. Jordan. Learning with mixtures of trees. *J. Mach. Learn. Res.*, 1:1–48, 2001.

[131] T. P. Minka. Variational bayes for mixture models: Reversing em. Technical report, MIT, 2000.

[132] L. A. Mirny and M. S. Gelfand. Using orthologous and paralogous proteins to identify specificity-determining residues in bacterial transcription factors. *J Mol Biol*, 321(1):7–20, Aug 2002.

[133] S. N., M. Lew, I. Cohen, A. Garg, and H. T.S. Emotion recognition using a cauchy naive bayes classifier. *Pattern Recognition, 2002. Proceedings. 16th International Conference on Publication Date*, 1:17–20, 2002.

[134] A. Nayeem, D. Sitkoff, and S. Krystek. A comparative study of available software for high-accuracy homology modeling: from sequence alignments to structural models. *Protein Sci.*, 15:808–824, Apr 2006.

[135] R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto, 1993.

[136] A. Nikseresht and M. Gelgon. Fast decentralized learning of a gaussian mixture model for large-scale multimedia retrieval. *Parallel, Distributed, and Network-Based Processing, 2006. PDP 2006. 14th Euromicro International Conference on*, pages 7 pp.–, 15-17 Feb. 2006.

[137] C. Ohmann, V. Moustakis, Q. Yang, and K. Lang. Evaluation of automatic knowledge acquisition techniques in the diagnosis of acute abdominal pain. *Artif Intell Med*, 8:23–36, Feb 1996.

[138] M. Ouyang, W. Welsh, and P. Georgopoulos. Gaussian mixture clustering and imputation of microarray data. *Bioinformatics*, 20:917–923, Apr 2004.

[139] L. Pachter and B. Sturmfels. *Algebraic Statistics for Computational Biology*. Cambridge University Press, New York, NY, USA, 2005.

[140] W. Pan, J. Lin, and C. Le. Model-based cluster analysis of microarray gene-expression data. *Genome Biol.*, 3:RESEARCH0009, 2002.

[141] K. Pearson. Mathematical contributions to the theory of evolution. *Proceedings of the Royal Society of London*, 64:163–167, 1898.

[142] J. Pei, W. Cai, L. N. Kinch, and N. V. Grishin. Prediction of functional specificity determinants from protein sequences using log-likelihood ratios. *Bioinformatics*, 22(2):164–171, Jan 2006. Evaluation Studies.

[143] W. D. Penny and D. P. Frost. Neural network modeling of the level of observation decision in an acute psychiatric ward. *Comput. Biomed. Res.*, 30(1):1–17, 1997.

[144] A. Pickles. Missing data, problems and solutions. *Kimberly Kempf-Leonard, ed., Encyclopedia of social measurement*, pages 689–694, 2005.

[145] D. Poole. Probabilistic horn abduction and bayesian networks. *Artificial Intelligence*, 64(1):81–129, 1993.

[146] M. Prastawa, E. Bullitt, N. Moon, K. Van Leemput, and G. Gerig. Automatic brain tumor segmentation by subject specific modification of atlas priors. *Acad Radiol*, 10:1341–1348, Dec 2003.

[147] J. Provost. Naive-bayes vs. rule-learning in classification of email. *Technical report, Dept. of Computer Sciences at the U. of Texas at Austin*, 1999.

[148] M. Ptashne. *A Genetic Switch: Gene Control and Phage l*. Cold Spring Harbor Laboratory Press, 2004.

[149] F. Qian, M. Li, L. Zhang, H.-J. Zhang, and B. Zhang. Gaussian mixture model for relevance feedback in image retrieval. *Multimedia and Expo, 2002. ICME '02. Proceedings. 2002 IEEE International Conference on*, 1:229–232 vol.1, 2002.

[150] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Readings in speech recognition*, pages 267–296, 1990.

[151] A. Raftery. Hypothesis testing and model selection via posterior simulation, 1995.

[152] C. E. Rasmussen. The infinite gaussian mixture model. In S. e. a. Solla, editor, *Advances in information processing systems 12*, pages 554–560. MIT Press, 2000.

[153] L. Rigouste, O. Cappé, and F. Yvon. Inference and evaluation of the multinomial mixture model for text clustering. *Inf. Process. Manage.*, 43(5):1260–1280, 2007.

[154] I. Rish. An empirical study of the naive bayes classifier. *IJCAI,Workshop on Empirical Methods in Artificial Intelligence*, 2001.

[155] C. P. Robert and G. Casella, editors. *Monte Carlo statistical methods*. Springer, New York, 2004.

[156] R. Rojas. *Neural networks: a systematic introduction*. Springer-Verlag New York, Inc., New York, NY, USA, 1996.

[157] K. Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of the IEEE*, 86(11):2210–2239, 1998.

[158] A. Sandelin, W. Alkema, P. Engstrom, W. W. Wasserman, and B. Lenhard. JASPAR: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Res*, 32(Database issue):91–94, Jan 2004.

[159] A. Sankar and V. R. R. Gadde. Parameter tying and gaussian clustering for faster, better, and smaller speech recognition.

[160] A. Schliep, I. G. Costa, C. Steinhoff, and A. Schönhuth. Analyzing gene expression time-courses. *IEEE/ACM Trans Comput Biol Bioinform*, 2(3):179–193, 2005.

[161] A. Schliep, C. Steinhoff, and A. Schönhuth. Robust inference of groups in gene expression time-courses using mixtures of hmm. *Proceedings of the ISMB 2004*, 2004.

[162] K. M. Schneider. Techniques for improving the performance of naive bayes for text classification. *Sixth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2005)*, pages 682–693, 2005.

[163] T. D. Schneider and R. M. Stephens. Sequence logos: A new way to display consensus sequences. *Nucleic Acids Res.*, 18:6097–6100, 1990.

[164] J. J. Schott, D. W. Benson, C. T. Basson, W. Pease, G. M. Silberbach, J. P. Moak, B. J. Maron, C. E. Seidman, and J. G. Seidman. Congenital heart disease caused by mutations in the transcription factor nkx2-5. *Science*, 281(5373):108–11, 1998.

[165] S. Schwartz, W. J. Kent, A. Smit, Z. Zhang, R. Baertsch, R. C. Hardison, D. Haussler, and W. Miller. Human-mouse alignments with BLASTZ. *Genome Res*, 13(1):103–107, Jan 2003.

[166] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464, 1978.

[167] A. Schönhuth, I. G. Costa, and A. Schliep. Semi-supervised clustering of yeast gene expression. In *Japanese-German Workshop on data analysis and classification*. Springer, 2006. Accepted.

[168] D. Seelow, R. Galli, S. Mebus, H. P. Sperling, H. Lehrach, and S. Sperling. d-matrix - database exploration, visualization and analysis. *BMC Bioinformatics*, 5:168, 2004. Journal Article Research Support, Non-U.S. Gov't England.

[169] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 and 623–656, July and October 1948.

[170] K. Sjolander, K. Karplus, M. Brown, R. Hughey, A. Krogh, I. S. Mian, and D. Haussler. Dirichlet mixtures: A method for improving detection of weak but significant protein sequence homology. Technical report, University of California at Santa Cruz, Santa Cruz, CA, USA, 1996.

[171] C. M. Smith, I. N. Shindyalov, S. Veretnik, M. Gribskov, S. S. Taylor, L. F. Ten Eyck, and P. E. Bourne. The protein kinase resource. *Trends Biochem Sci*, 22(11):444–446, Nov 1997.

[172] J. E. Smith, S. Holtzman, and J. E. Matheson. Structuring conditional relationships in influence diagrams. *Oper. Res.*, 41(2):280–297, 1993.

[173] Q. Song and M. Shepperd. Missing data imputation techniques. *Int. J. Bus. Intell. Data Min.*, 2(3):261–291, 2007.

[174] S. Sperling, C. H. Grimm, I. Dunkel, S. Mebus, H. P. Sperling, A. Ebner, R. Galli, H. Lehrach, C. Fusch, F. Berger, and S. Hammer. Identification and functional analysis of cited2 mutations in patients with congenital heart defects. *Hum Mutat*, 26(6):575–82, 2005. 1098-1004 (Electronic) Journal Article.

[175] R. Staden. Computer methods to locate signals in nucleic acid sequences. *Nucleic Acids Res*, 12(1 Pt 2):505–519, Jan 1984.

[176] R. Staden. Methods for calculating the probabilities of finding patterns in sequences. *Comput Appl Biosci*, 5(2):89–96, Apr 1989.

[177] T. A. Stephenson. Conditional Gaussian mixtures. IDIAP-RR 11, IDIAP, 2003.

[178] A. Stolcke and S. M. Omohundro. Best-first model merging for hidden Markov model induction. Technical Report TR-94-003, University of California at Berkeley, 1947 Center Street, Berkeley, CA, 1994.

[179] G. D. Stormo. Consensus patterns in DNA. *Methods Enzymol*, 183:211–221, 1990.

[180] G. D. Stormo. DNA binding sites: representation and discovery. *Bioinformatics*, 16(1):16–23, Jan 2000. Historical Article.

[181] T. Strachan and A. P. Read. *Human Molecular Genetics 2*. Wiley-Liss, 1999.

[182] J. Söding. Protein homology detection by HMM-HMM comparison. *Bioinformatics*, 21:951–960, Apr 2005.

[183] S. Takahashi and S. Sagayama. Tied-structure hmm based on parameter correlation for efficient model training. In *ICASSP '96: Proceedings of the Acoustics, Speech, and Signal Processing, 1996. on Conference Proceedings., 1996 IEEE International Conference*, pages 467–470, Washington, DC, USA, 1996. IEEE Computer Society.

[184] J. Thomas, J. Touchman, R. Blakesley, G. Bouffard, S. Beckstrom-Sternberg, E. Margulies, M. Blanchette, A. Siepel, P. Thomas, J. McDowell, B. Maskeri, N. Hansen, M. Schwartz, R. Weber, W. Kent, D. Karolchik, T. Bruen, R. Bevan, D. Cutler, S. Schwartz, L. Elnitski, J. Idol, A. Prasad, S. Lee-Lin, V. Maduro, T. Summers, M. Portnoy, N. Dietrich, N. Akhter, K. Ayele, B. Benjamin, K. Cariaga, C. Brinkley, S. Brooks, S. Granite, X. Guan, J. Gupta, P. Haghighi, S. Ho, M. Huang, E. Karlins, P. Laric, R. Legaspi, M. Lim, Q. Maduro, C. Masiello, S. Mastrian, J. McCloskey, R. Pearson, S. Stantripop, E. Tiongson, J. Tran, C. Tsurgeon, J. Vogt, M. Walker, K. Wetherby, L. Wiggins, A. Young, L. Zhang, K. Osoegawa, B. Zhu, B. Zhao, C. Shu, P. De Jong, C. Lawrence, A. Smit, A. Chakravarti, D. Haussler, P. Green, W. Miller, and E. Green. Comparative analyses of multi-species sequences from targeted genomic regions. *Nature*, 424(6950):788–793, Aug 2003.

[185] W. Thompson, M. J. Palumbo, W. W. Wasserman, J. S. Liu, and C. E. Lawrence. Decoding human regulatory circuits. *Genome Res*, 14(10A):1967–1974, Oct 2004.

[186] B. Todd and R. Stamper. The relative accuracy of a variety of medical diagnostic programs. *Methods Inf Med*, 33:402–416, Oct 1994.

[187] M. Toenjes, M. Schueler, S. Hammer, U. Pape, J. Fischer, F. Berger, M. Vingron, and S. Sperling. Prediction of cardiac transcription networks based on molecular data and complex clinical phenotypes. *Mol Biosyst*, 4:589–598, Jun 2008.

[188] A. Tomovic and E. Oakeley. Position dependencies in transcription factor binding sites. *Bioinformatics*, 23:933–941, Apr 2007.

[189] A. Tong, B. Drees, G. Nardelli, G. Bader, B. Brannetti, L. Castagnoli, M. Evangelista, S. Ferracuti, B. Nelson, S. Paoluzi, M. Quondam, A. Zucconi, C. Hogue, S. Fields, C. Boone, and G. Cesareni. A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules. *Science*, 295:321–324, Jan 2002.

[190] J. J. Verbeek, N. Vlassis, and J. R. J. Nunnink. A variational em algorithm for large-scale mixture modeling. In *In Proc. 8th Ann. Conf. of the Advanced School for Computing and Imaging (ASCI 2003), Het Heijderbos, Heijen*, 1998.

[191] K. L. Wang and J. R. Warner. Positive and negative autoregulation of REB1 transcription in Saccharomyces cerevisiae. *Mol Cell Biol*, 18(7):4368–4376, Jul 1998.

[192] S. M. Ware, J. Peng, L. Zhu, S. Fernbach, S. Colicos, B. Casey, J. Towbin, and J. W. Belmont. Identification and functional analysis of zic3 mutations in heterotaxy and related congenital heart defects. *Am J Hum Genet*, 74(1):93–105, 2004.

[193] G. A. Weiss, C. K. Watanabe, A. Zhong, A. Goddard, and S. S. Sidhu. Rapid mapping of protein functional epitopes by combinatorial alanine scanning. *Proceedings of the National Academy of Sciences of the United States of America*, 97(16):8950–8954, 2000.

[194] T. Werner. Models for prediction and recognition of eukaryotic promoters. *Mamm Genome*, 10(2):168–175, Feb 1999.

[195] N. Wicker, G. R. Perrin, J. C. Thierry, and O. Poch. Secator: a program for inferring protein subfamilies from phylogenetic trees. *Mol Biol Evol*, 18(8):1435–1441, Aug 2001.

[196] J. R. Williams, C. Thayyullathil, and N. E. Freitag. Sequence variations within PrfA DNA binding sites and effects on Listeria monocytogenes virulence gene expression. *J Bacteriol*, 182(3):837–841, Feb 2000.

[197] M. P. Windham and A. Cutler. Information ratios for validating mixture analyses. *Journal of the American Statistical Association*, 87(420):1188–1192, 1992.

[198] E. Wingender, P. Dietze, H. Karas, and R. Knüppel. TRANSFAC: a database on transcription factors and their DNA binding sites. *Nucleic Acids Res.*, 24:238–241, Jan 1996.

[199] S. A. Wolfe, H. A. Greisman, E. I. Ramm, and C. O. Pabo. Analysis of zinc fingers optimized via phage display: evaluating the utility of a recognition code. *J Mol Biol*, 285(5):1917–1934, Feb 1999.

[200] G. Yu, B.-H. Park, P. Chandramohan, R. Munavalli, A. Geist, and N. F. Samatova. In silico discovery of enzyme-substrate specificity-determining residue clusters. *J Mol Biol*, 352(5):1105–1117, Oct 2005.

[201] R. Zee, N. Cook, S. Cheng, H. Erlich, K. Lindpaintner, R. Lee, and P. Ridker. Threonine for alanine substitution in the eotaxin (CCL11) gene and the risk of incident myocardial infarction. *Atherosclerosis*, 175:91–94, Jul 2004.

[202] G. Zhang, Y. Liu, A. E. Ruoho, and J. H. Hurley. Structure of the adenylyl cyclase catalytic core. *Nature*, 386(6622):247–253, Mar 1997.

Appendix A

Notation

C	Component indicator variable
D	Data set of N realizations of X.
$E[X]$	Expectation of random variable X
G	CSI structure of a mixture model
g_j	CSI structure for feature X_j
$g_j(k)$	Mapping from component indices to CSI group indices, i.e. $g_j(k) : (1, ..., K) \rightarrow (1, ..., Z_j)$
g_{jr}	r'th group in CSI structure g_j
γ	Hyper–parameter of the prior over the number of components $P(K)$
H	Hidden data, i.e. labels which assign samples in D to components
K	Number of components of a mixture.
μ_{kj}	Mean parameter of a Gaussian distribution at index j in component k
N	Number of samples in D.
ω	Hyper–parameter of the prior over the CSI structure $P(G)$
p	Number of dimensions in X.
$P(x_i\|\Theta)$	Probability density or mass function over variable X parameterized by Θ
ϕ_{kj}	Discrete distribution over M–symbol alphabet Σ at index j in component k
ϕ_{kjs}	s'th entry of discrete distribution ϕ_{kj}
π	Mixture weights, K–dimensional stochastic vector
Σ	Alphabet with M symbols
σ^2_{kj}	Variance parameter of a Gaussian distribution at index j in component k
τ_{ki}	Posterior of component membership for sample i and component k

Θ	Mixture model parameterization
θ_k	Parameterization of the k'th component of a mixture
θ_{kj}	Parameters for the j'th distribution in naïve Bayes component k
X	Multivariate random variable with p dimensions
x_i	Realization of X, i.e. vector of length p
x_{ij}	Realization of X_j, j'th element of x_i
X_j	Random variable, j'th element of X
Z_j	Number of groups in CSI structure g_j

Appendix B

Abbreviations

AIC	*Akaike information criterion*
AoArch	Aortic arch
AoAsc	Aorta ascendensis
AoV	Aortic valve
AV	Atrioventricular
BIC	*Bayesian information criterion*
BN	Bayesian network
CPT	*conditional probability table*
CSI	*context-specific* independence
DMP	Dirichlet mixture prior
DNA	deoxyribonucleic acid DNA
DORV	*double outlet right ventricle*
EM	Expectation Maximization
FN	false negatives
FP	false positives
HMM	Hidden Markov model
IAS	Interatrial septal defect
IVS	interventicluar septum
LA	left atrium
LDH	lactate dehydrogenases
LV	left ventricle
MAP	*maximum a posteriori*
MCMC	Markov chain Monte Carlo
MDH	malate dehydrogenases
mRNA	*messenger* ribonucleic acid
ML	maximum likelihood
MPA	main pulmonary artery
MSA	multiple sequence alignment
NEC	*Normalized Entropy criterion*
PDB	*Protein Data Bank*
PFO	Patent foramen ovale

PV	pulmonary valve
PV	pulmonary valve
PWM	positional weight matrix
RA	right atrium
RV	random variables
RV	right ventricle
TF	transcription factors
TFBS	transcription factor binding sites
TK	tyrosine kinases
TN	true negatives
TOF	*tetralogy of Fallot*
TP	true positives
sEM	structural EM

Appendix C

Nucleotide & Amino Acid Codes

Nomenclature for the four nucleotides and 20 standard amino acids is determined by the *International Union of Pure and Applied Chemistry* (IUPAC) codes.

Nucleotide	1-letter code
adenine	A
cytosine	C
guanine	G
thymine	T

Table C.1: *IUPAC codes for the four nucleotides.*

Amino acid	3-letter code	1-letter code
Alanine	Ala	A
Arginine	Arg	R
Asparagine	Asn	N
Aspartic acid	Asp	D
Cysteine	Cys	C
Glutamic acid	Glu	E
Glutamine	Gln	Q
Glycine	Gly	G
Histidine	His	H
Isoleucine	Ile	I
Leucine	Leu	L
Lysine	Lys	K
Methionine	Met	M
Phenylalanine	Phe	F
Proline	Pro	P
Serine	Ser	S
Threonine	Thr	T
Tryptophan	Trp	W
Tyrosine	Tyr	Y
Valine	Val	V

Table C.2: *IUPAC codes for the twenty standard amino acids.*

Appendix D

Random CSI Models

When generating random CSI models, some care has to be taken that the structure and model parameters are not inconsistent in the sense that the structure gives separate groups to two components, whereas the randomly selected parameters, by chance, are very similar. In order to circumvent the problem, an exhaustive structure learning is performed to after the random sampling of parameters to make sure this situation does not occur.

Therefore, the protocol to generate a data set D from a random CSI model with given dimension p, numbers of component K and number of samples N is

- sample structure G uniformly from all possible structures given p and K
- for all features X_j sample parameters $\theta_{X_j|g_{jr}}$ according to the local structures g_j
  - if $\theta_{X_j|g_{jr}}$ is discrete, sample parameters from a uniform Dirichlet distribution with $M = 8$
  - if $\theta_{X_j|g_{jr}}$ is Gaussian, sample μ and σ^2 from uniform distributions $U(-25.0, 25.0)$ and $U(0.3, 5.0)$ respectively
- perform structure learning with exhaustive enumeration to ensure consistency
- sample N data points to obtain the data set D

Appendix E

Zusammenfassung

Die Dissertation beschäftigt sich mit der Analyse von biologischen Daten aus dem Bereich Genetik und Molekularbiologie. Der Fokus der Arbeit liegt auf dem 'Clustering', d.h. der automatischen Unterteilung eines Datensatzes in Gruppen von ähnlichen Dateneinträgen. Diese Gruppen werden dann im Bezug auf ihre unterschiedliche biologische Bedeutung analysisert.

Der statistische Formalismus, der in der Arbeit angewendet wird, ist das Mischmodell. Mischmodelle weisen eine Reihe von erstrebenswerten Eigenschaften auf. Sie sind flexibel in der Abbildung verschiedener Datensätze, erlauben effiziente Parameterschätzung und sind robust gegenüber verrauschten Daten. Im Kapitel 2 werden der Mischmodellformalismus, der Algorithmus zur Parameterschätzung und Details zur Anwendung für das Clustering beschrieben. Im Kapitel 3 wird eine neuartige Erweiterung der konventionellen Mischmodelle, die *kontext-spezifische Unabhängigkeit* (eng. CSI) motiviert und eingeführt. Die CSI Erweiterung erlaubt die automatische Anpassung der Modell-Komplexität an die Variabilität eines gegebenen Datensatzes. Dies hat nicht nur den Vorteil, dass nur so viele Parameter geschätzt werden müssen, wie benötigt sind, um den Datensatz abzubilden. Die gelernte CSI Struktur erlaubt auch die Charakterisierung der vom Clustering erzeugten Gruppen. Im Kapitel 4 wird der Algorithmus, der zur Parameterschätzung von CSI-Mischmodellen benötigt wird, eingeführt und getestet.

Die erste Anwendung, die in der Arbeit behandelt wird, ist die Modellierung von Transkriptionsfaktorenbindestellen (TFBS) mit Hilfe von CSI-Mischmodellen (Kapitel 5). Klassicherweise wird das Bindeverhalten eines Transkriptionsfaktors (TF) mit einer einfachen Positionsgewichtsmatrix (enlg. PWM) modelliert. In dieser Studie zeigen wir, dass für TFs, die mehrere, unterschiedliche Bindemotive aufweisen, CSI-Mischmodelle die geeignete Modellklasse darstellen. Am Beispiel des TFs Leu3 konnten wir zeigen, dass die Anwendung von Mischmodellen biologisch motiviert ist. Desweiteren fanden wir in einer Sequenzkonservierungsstudie zwischen Mensch und Maus für einen Datensatz von 64 TFs, dass CSI-Mischmodelle durchgehend bessere Ergebnisse liefern als konventionelle Mischmodelle.

Die zweite Anwendung beschäftigt sich mit dem Clustering von Proteinsequenzen aus funktionell verwandten Proteinunterfamilien (Kapitel 6). Es ist bekannt, dass viele solcher

Unterfamilien unterschiedliche Substratspezifitäten aufweisen. Oftmals beruht die unterschiedliche Spezifität nur auf einer kleinen Anzahl an Aminosäureresten. Das Clustering eines Datensatzes von Proteinsequenzen umfasst dann nicht nur die Einteilung in Untergruppen, sondern auch die Vorhersage der funktionellen Reste, das heißt, der Positionen im Protein, die die Substratspezifität bestimmen. Um die CSI-Mischmodelle für diesen Zweck anzuwenden, führen wir eine neuartige Modellerweiterung, basierend auf der Dirichletverteilung ein. In der Folge analyisieren wir die Performanz des Ansatzes auf einer Reihe von Proteinsequenzdatensätzen.

Die dritte Anwendung befasst sich mit dem Clustering von Phänotypen von Patienten mit angeborenen Herzfehlern (Kapitel 7). Der Datensatz umfasst 26 binäre Phänotypen, die jeweils die An- oder Abwesenheit von verschiedenen anatomischen Missbildungen des Herzens repräsentieren. Die Anwendung von CSI-Mischmodellen für das Clustering eines Datensatzes von 65 Herzpatienten führte zur Unterteilung in vier Untergruppen. Die Analyse der Untergruppen zeigte sowohl eine gute Korrelation mit klassischen Herzkrankheitstypen, als auch eine neuartige Gruppierung, die eine Mischform von klassisch beschriebenen Herzkrankheiten ist.

Ehrenwörtliche Erklärung

Hiermit erkläre ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel und Quellen verwendet habe.

Berlin, 12.03.2009 Benjamin Georgi