A PAC-Bayesian Approach to Structure Learning

Thesis submitted for the degree of "Doctor of Philosophy"
by

Yevgeny Seldin

Submitted to
the Senate of the Hebrew University of Jerusalem
September 2009

This work was carried out under the supervision of
Prof. Naftali Tishby

Abstract

Structure learning is an important sub-domain of machine learning. Its goal is a high level understanding of the data. For example, given an image as a collection of pixels, the goal is to identify the objects present in the image, such as people, trees, birds, to infer their actions (e.g., standing, flying) and interactions (e.g. a man is feeding a bird). Structure learning has multiple applications in a variety of domains, including the analysis of cellular processes in biology, computer vision, and natural language processing, to name only a few. In the recent couple of decades enormous progress has been made in data analysis methods that are not based on explicit structure, such as Support Vector Machines (SVMs) or other kernel-based methods. Nevertheless, the importance of learning explicit structures remains crucial. There are multiple benefits to structure-based learning as opposed to methods, where structure is present implicitly, e.g. in the form of a kernel. The primary reason is that it is easier for us as humans to manipulate objects like cars and people rather than raw pixels or kernel matrices. For example, a natural query for a human would be: "find the photos of me picking mushrooms last summer". Thus we need a learning algorithm to go all the way up from pixels to persons, mushrooms, trees, etc., and also actions like picking, standing, and so on. Another important reason to learn structure in data is "to understand the world around us" for example, to understand biological processes or even how our own brain works. It also becomes easier to control or influence different processes once we learn their structure and extract simple rules governing their dynamics.

Often, the amount of supervision available for a learning algorithm in a

structure learning task is limited or even non-existent. Even when present, supervision is often at a high level, whereas the data are represented at a low level. For example, we may infer from an image label that it is an image of a cow, but the algorithm still has to infer the cow's location and shape. However, many studies have shown that even completely unsupervised learning methods are able to identify meaningful structures present in data and can facilitate high level decisions. But despite their remarkable success in practice, our conceptual understanding of structure learning approaches is highly limited. The issue is so basic that even if we are given two reasonable solutions to some problem (for example, two possible segmentations of an image) we are unable to make a well-founded judgment as to which one is better. Typical forms of evaluation are quite subjective, such as "this segmentation looks more natural" or "this has a higher level of correlation with human annotation". However, this form of evaluation is hard to apply in domains where our own intuition is limited, such as bioinformatics or neuroscience. Model order selection and a proper choice of evaluation procedures have remained open questions in structure learning for over half a century. The lack of solid theoretical foundations has given rise to multiple heuristic approaches which are hard to compare and in the end are slowing down the development of the whole field.

The picture is completely different in supervised learning. The first advantage of supervised learning is that it has a well-defined learning objective - the prediction of a label. From this point it becomes clear how to conduct a formal analysis of different learning approaches (usually in the form of a derivation of a generalization or sample complexity bounds) and how to evaluate them. Nowadays, most successful classification algorithms are accompanied by generalization guarantees and many were derived as algorithms for optimization of generalization guarantees. The existence of a clear objective and generalization guarantees for most algorithms makes it possible to compare solutions to the same problem obtained by different approaches (e.g., SVMs and decision trees) both theoretically and practically. The ability to make a formal analysis and compare different approaches accounts for the rapid rise in supervised learning in recent decades.

In this thesis it is claimed that the ill-posed nature of unsupervised learning approaches and in particular unsupervised learning approaches to structure learning is caused by the fact that unsupervised learning problems are usually taken out of context. Here we argue that one does not learn structure just for the sake of learning structure, but rather in order to facilitate solving some higher level task. By identifying that task and looking at structure learning from the point of view of its utility in solving the higher level task we return the structure learning problem to its context. This enables a clear and objective comparison of different approaches to structure learning, similar to the way it is done in the supervised learning. We can also examine in which situations knowing structure is beneficial to solving a task, or whether unstructured methods such as SVMs or Principal Component Analysis (PCA) would perform better. The problem of model order selection can be answered naturally in this context. It also improves our understanding of which questions the structure we have learned can answer and which it cannot. Thus, it is not only desirable, but necessary to consider structure learning within the wider context of its possible applications.

We demonstrate our approach to the formulation of structure learning within the context of a higher level task using the example of co-clustering. Co-clustering is a widely used approach to the analysis of data matrices by simultaneous grouping (clustering) of "similar" rows and columns of a data matrix; for instance, clustering of viewers and movies in collaborative filtering, genes and conditions in gene expression data analysis, or words and documents in text analysis. Co-clustering was traditionally considered an unsupervised approach to data analysis. Various solutions were designed by approximating different properties of the data at hand, but in most cases there was no way to compare different solutions and perform model order selection. This dissertation examines co-clustering solutions in the context of their ability to predict new events generated by the same distribution that generated the data matrix. Within this context it becomes possible to carry out generalization analysis of co-clustering, model order selection, and to compare co-clustering with other approaches to solving this problem.

We find it convenient to use a PAC-Bayesian framework to carry out a

formal analysis of generalization properties of co-clustering. This work also makes several contributions to the domain of PAC-Bayesian generalization bounds, which up to now have solely been used in the context of generalization analysis of classification approaches. Here we derive PAC-Bayesian generalization bound for density estimation. We also show that the obtained bounds for co-clustering are actually generalization bounds for a particular form of graphical models, where each hidden variable is connected to a single observable variable. The bounds suggest that the generalization power of such graphical models is optimized by a tradeoff between empirical performance and mutual information preserved by the hidden variables on the observed variables. The regularization of the models by the mutual information comes as a result of the use of combinatorial priors in PAC-Bayesian bounds and is yet another contribution of our work. Our combinatorial priors can be compared to Gaussian and Laplacian priors that were used in other works on PAC-Bayesian bounds and result in L_2-norm and L_1-norm regularization, respectively.

In the applications chapter we show that the tradeoff between empirical performance and mutual information preserved on the observed variables that was derived from the bounds enables a complete control over regularization of the co-clustering model. It further achieves state-of-the-art results on the MovieLens collaborative filtering dataset.

In an excursus, we show that the bounds developed for co-clustering can be applied in classifications by a single parameter (e.g., prediction of a probability of a disease based on country visited). In the applications chapter we demonstrate that the bounds are especially tight in this task and are less than 15% away from the test error. We suggest that the bounds can be applied to feature ranking. We futher show that such an approach to feature ranking, where features are ranked by their generalization potential, is much more successful than feature ranking based on empirical mutual information or normalized correlation with the label.

As a continuation of the main thrust of the thesis, we show that it is possible to extend the analysis of co-clustering to more complex domains. Specifically, we show that the same kind of a tradeoff between empirical

performance and mutual information that the hidden variables preserve on the observed variables also holds in graphical models, where hidden variables are connected in a tree shape and the observed variables are at the leaves of the tree. We also demonstrate that PAC-Bayesian bounds are able to exploit the factor form of graphical models and make it possible to derive bounds that depend on the sizes of the cliques of the graph rather than the size of the parameter space. We further suggest viewing the problem of learning graphical models from the point of view of the ability of the graphical model to predict new events generated by the same distribution rather than the frequently applied practice of fitting the train set. Extensions of the results to more complex graphical models as well as the development of efficient algorithms for optimization of structures of graphical models with respect to the bounds are subjects for future research.

Contents

Chapter 1

Introduction

Structure learning is a long-standing problem that has attracted extensive interest in fields such as bioinformatics, image processing, neuroscience, and so on. In many situations a learning task must take place with very limited or even no supervision. Even when supervision is available it is often given at a high level, whereas the data are represented at a low level; thus we have no guidance to make it all the way up from the low level to the high level. Nevertheless, many studies including our own in biosequence analysis [79, 13, 78, 77] and image processing [81] have shown that even completely unsupervised learning methods are able to identify meaningful structures present in the data and can facilitate high level decisions. However, the conceptual understanding of unsupervised learning approaches to structure learning is far from satisfactory. This is true to such an extent that even if we are given two reasonable solutions to some problem we are unable to provide a well-founded judgment as to which one is better. We further illustrate this in the following motivation example based on [81].

1.1 Motivation Example

In [81] we presented a method for unsupervised content based clustering of collections of images. Clustering of image collections into meaningful classes has many applications in the organization of image databases and the design

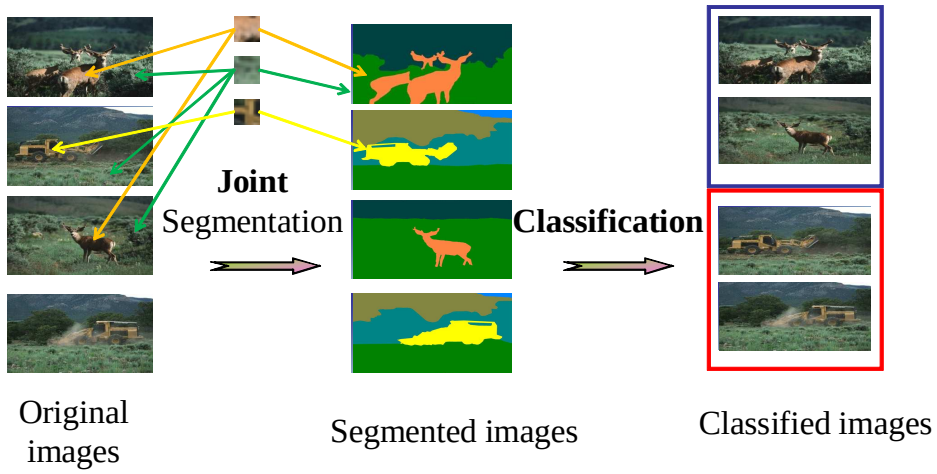


Figure 1.1: **[81]: Unsupervised Clustering of Images using their Joint Segmentation - workflow illustration.** See text for details. The figure is for illustration purposes only, the actual outcome of the algorithm is presented in Figures 1.2-1.5 and in more detail in [81].

of human interfaces for collection browsing. In [81] we suggested a two-level approach to solve this task, depicted in Figure 1.1. The first level involves learning a centroid-based mixture of textures model to represent all the images in the collection. The idea behind this step is that if there is an object common to multiple images, the corresponding texture (or textures) should appear in those images. Through learning the mixture of textures model each image is represented with a small number of representative textures. The small number of representative textures is very likely to correspond to objects that appeared in multiple images. Thus, we obtained a simple representation of each image in terms of objects common to multiple images, so called "visual words". The second level involved drawing an analogy between textures-words and images-documents and applying algorithms from the field of unsupervised document classification to cluster the images.

Some results obtained by the above approach are shown in Figures 1.2-1.5 and we refer the reader to the original paper for more details. We point out that although the results look nice, there are many open questions that remain unresolved. One of the most important is how we can formally

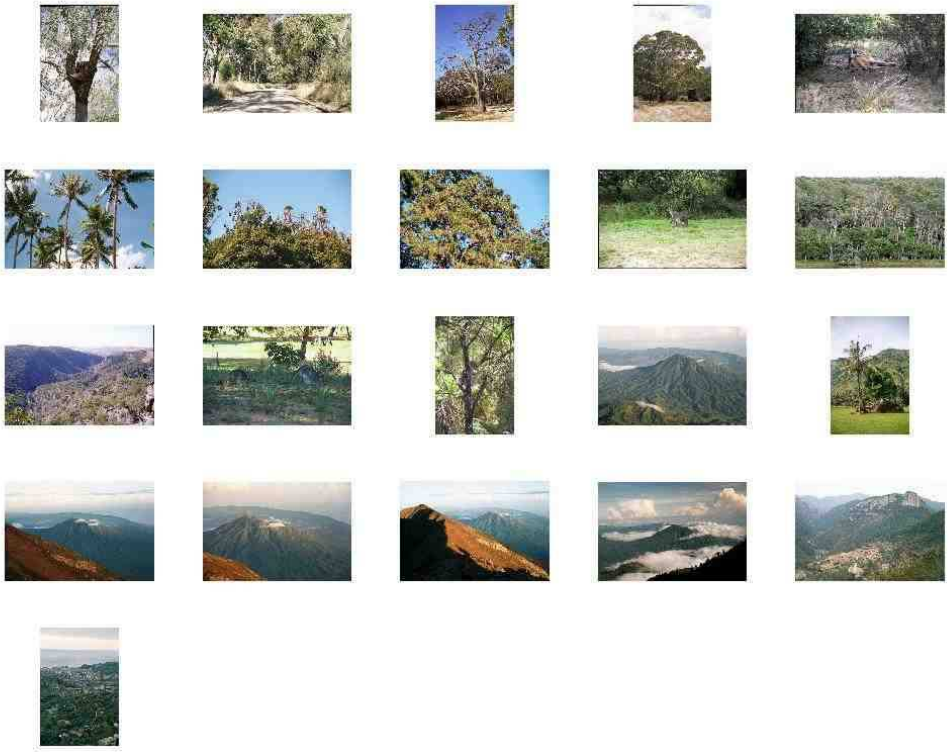
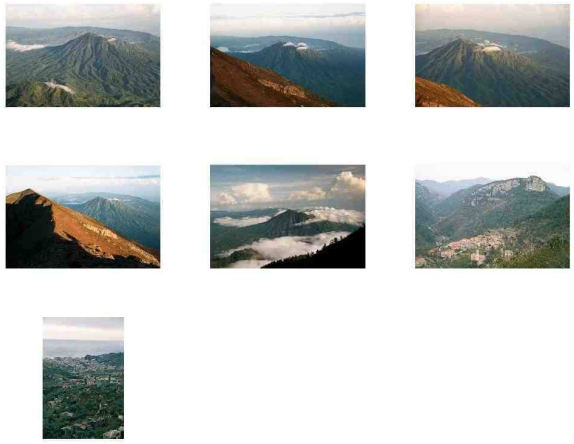


Figure 1.2: **Unsupervised Clustering of a Collection of Images from Australia. From [81].** The original dataset.

say which solution is better. On the most superficial level this becomes the question of whether we should partition the collection into two, three, four or some other number of clusters. However it should be recalled that we applied a two-level unsupervised approach, which thus raised a similar question of how many textures in the mixture of textures model should have been selected at the lower level. And there are multiple ways to model the textures aside from other completely different ways to approach this task that could be chosen. In the absence of a clear evaluation procedure we get lost in an ocean of heuristic approaches.

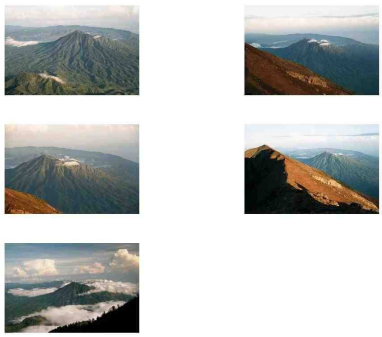
It should be borne in mind that the questions of model order selection and comparison of different approaches are open questions that apply even to the simplest and the oldest problem of learning a one-level mixture model.



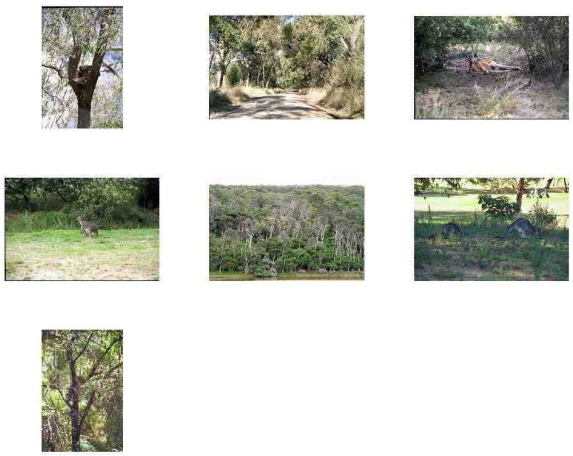
(a) Cluster 1 out of 2

(b) Cluster 2 out of 2

Figure 1.3: **Unsupervised Clustering of a Collection of Images from Australia. From [81].** Clustering of the dataset into two clusters.



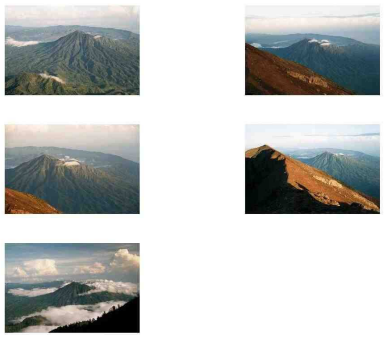
(a) Cluster 1 out of 3



(b) Cluster 2 out of 3

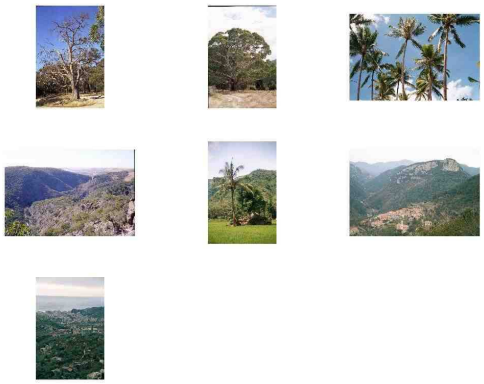
(c) Cluster 3 out of 3

Figure 1.4: **Unsupervised Clustering of a Collection of Images from Australia. From [81].** Clustering of the dataset into three clusters.



(a) Cluster 1 out of 4

(b) Cluster 2 out of 4



(c) Cluster 3 out of 4

(d) Cluster 4 out of 4

Figure 1.5: **Unsupervised Clustering of a Collection of Images from Australia. From [81].** Clustering of the dataset into four clusters.

The major message behind the above example was to underline that an absence of a good answer to these questions creates even harder problems once we advance to models with more than one level. Such models were demonstrated to have better performance in practice in multiple recent works, of which [16, 36, 42] are just a few examples. However, no formal statements supporting these approaches have been proposed yet.

The reader is warned that this thesis will not provide an answer to the complex problem presented in this motivation example. However it does suggest some formulations and analyses that constitute substantial steps in the direction of the high-level goal.

1.2 Outline

This thesis is divided into four parts. The first part, consisting of Chapter 2, is more philosophical and presents our view on how to formulate learning questions. Here we define out-of-sample performance of unsupervised learning problems. We further point out that there are two types of semantically different problems that are usually approached within the same framework with co-clustering and define out-of-sample performance for each of the two settings. The second part, consisting of chapters 3 to 5, is the mathematical part of this work. In Chapter 3 we review and extend PAC-Bayesian generalization bounds. In Chapter 4 we apply PAC-Bayesian bounds to perform generalization analysis of co-clustering. In Chapter 5 we suggest how to extend the analysis of co-clustering to more complex graphical models. The third part, consisting of chapter 6, is algorithmic and presents algorithms for minimization of the bounds developed in the second part. The last part, consisting of Chapter 7, presents some applications to real life data. In particular, using our algorithm for co-clustering we achieve state-of-the-art performance in rating predictions in the MovieLens dataset. Chapter 8 discusses the results of this thesis and suggests some directions for future research.

1.3 Summary of Main Contributions

The main contributions of this thesis can be summarized as follows.

- To the best of our knowledge this is the first time that out-of-sample performance of co-occurrence data analysis has been defined and analyzed. Highlighting the difference between co-occurrence data and functional data (such as in collaborative filtering) was also an important step in proper analysis of both settings.

- Another contribution has to do with generalization bounds for discriminative prediction and density estimation based on co-clustering. This enables both theoretical and practical comparison of the co-clustering approach to data analysis with completely different approaches such as Probabilistic Latent Semantics Analysis and various forms of matrix factorization.

- Several important extensions to the PAC-Bayesian bounds were derived in this dissertation. One is an application of the PAC-Bayesian framework to derive generalization bounds for discrete density estimation.

- Another important extension of PAC-Bayesian bounds is the introduction of combinatorial priors. Combinatorial priors yield regularization terms in the form of mutual information and are more appropriate to combinatorial hypothesis spaces, as opposed to L_2-norm normalization resulting from Gaussian priors or L_1-norm normalization corresponding to Laplacian priors.

- It is shown here that the application of PAC-Bayesian bounds to co-clustering can be extended to more general graphical models. In particular, generalization bounds for directed graphical models in a tree shape and their moralized undirected counterparts are derived.

- Algorithms for minimization of the bounds for co-clustering are suggested.

- The application of minimization algorithms to the MovieLens dataset resulted in state-of-the-art performance.
- As an excursus, it is shown that the bounds for co-clustering can be reduced to obtain bounds for classification by a single feature. It is further demonstrated that the bounds can be applied to feature ranking and that such feature ranking is superior to feature ranking by empirical mutual information or correlation with the label.

Part I

Formulation

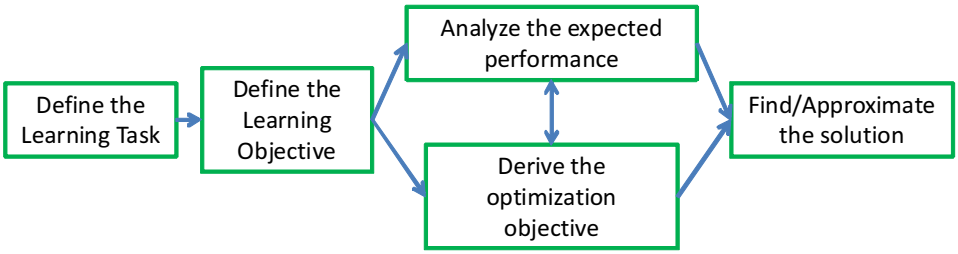
Chapter 2

How to Formulate a Learning Problem

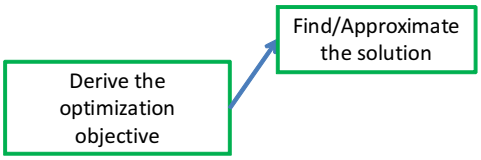
Any good study should provide a clear definition of the question it is trying to answer. It is claimed here that the current state of disarray in the field of unsupervised learning is the result of the absence of a clear formulation of a question that unsupervised learning is trying to answer. For that reason, the first technical chapter of this work is devoted to rethinking the objectives in unsupervised learning and structure learning. As a guide line it takes the example of the much better developed field of supervised learning. We argue that the beauty and clarity of supervised learning starts with a clear definition of the learning objective. In this chapter we try to define its analog for unsupervised learning.

2.1 Supervised vs. Unsupervised Workflow

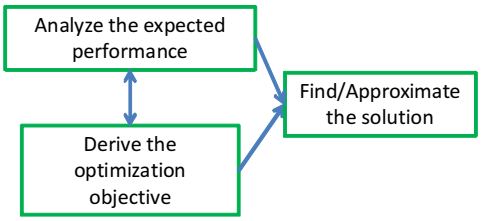
The process of formulation and solution of most supervised learning problems, including online and batch learning [33, 31, 104, 22], can be mapped onto the diagram in Figure 2.1.a. It starts with a definition of a learning task. This is expressed in the selection of a label, which is the property we want the learning algorithm to predict. Next, we select our learning objective. This is expressed in the selection of the way we will measure the



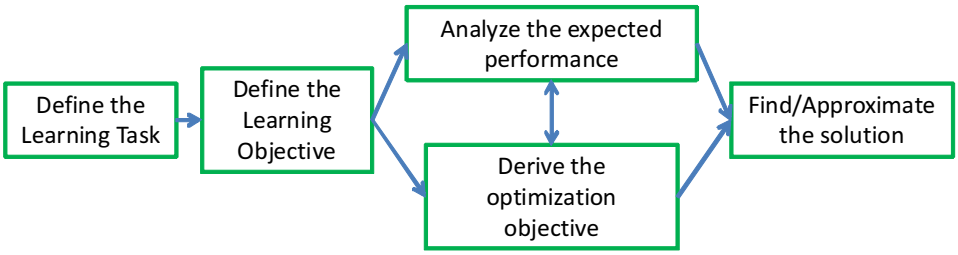
(a) Supervised Learning



(b) "Classical" Unsupervised Learning



(c) Unsupervised Learning with Stability Analysis



(d) Unsupervised Learning - Desired Framework

Figure 2.1: Typical framework for analysis of (a) supervised learning problems, (b) unsupervised learning problems, (c) analysis of unsupervised learning based on stability, and (c) the desired framework for analysis of unsupervised learning. See text for details.

errors in our predictions. Examples include zero-one loss, log-loss, quadratic loss, and so on. This also incorporates the choice of whether we would like to minimize the expected error, or the maximal error, the regret (in online learning), etc. In some situations the selected objective is computationally intractable and we are forced to replace it with a tractable relaxation. This includes the example of replacement of the zero-one loss with hinge loss in SVMs. In most cases it is possible to conduct a formal analysis and provide guarantees (usually in the form of generalization bounds or regret bounds in online learning) on the expected value of the objective for the solution. Nowadays, most of successful classification algorithms are accompanied by such generalization guarantees and many were derived as algorithms for optimization of the guaranteed generalization performance. Finally, the original or relaxed version of the learning objective is solved or approximated. The existence of the theoretical analysis and the formal guarantees enable theoretical comparison of completely different and unrelated approaches to solving the same problem, such as SVMs and decision trees, for example. This was the impetus for the development of supervised learning in recent decades. However all this was possible because there was a well-defined learning task.

Most unsupervised learning approaches start directly from the formulation of the learning functional, as shown in Figure 2.1.b. Possibly the most famous example is the k-means objective, where we want to find k centroids, so that the average distance of points to the nearest centroids $\sum_i \min_{\{c_j\}_{j=1}^k} \|x_i - c_j\|_2$ is minimized [33]. This is far from being the only formulation of the objective and multiple alternative formulations exist [47, 106], but it is impossible to compare solutions that optimize different optimization objectives, since the objectives are unrelated. It is also not clear how to perform model order selection; namely, how to compare a clustering solution with two clusters to a clustering solution with three clusters.

This problem, especially in the context of clustering, has troubled multiple researchers [105]. In recent years extensive attempts have been made to address the question of model order selection in clustering from the point of

view of its stability [15, 85, 86, 87, 14, 58]. This approach provides an external analysis of optimization solutions of a given objective and is depicted in Figure 2.1.c. Although it was proved that in a large sample regime stability can be used for model order selection [87], no lower bounds on the minimal sample size required for stability estimates to hold can be proved, and in fact in certain cases stability indices based on arbitrarily large samples can be misleading [14]. Since in any practical situation the amount of data available is limited, currently existing stability indices cannot be used for reliable model order selection. Furthermore it is not clear whether stability indices can be used to compare between solutions based on different optimization objectives.

This leads to one of the major points explored in this dissertation. We argue that one does not learn structure (e.g., clustering) just for the sake of learning structure, but rather in order to solve some higher level task. Thus, we should evaluate structures in terms of their utility in solving a given high-level task. This is expressed in the diagram in Figure 2.1.d. In section 2.2 we show how to map the suggested workflow on a real problem of co-clustering.

The idea to consider structure learning in the context of a higher level task was inspired by the Information Bottleneck (IB) principle [99, 90, 92, 84]. The IB principle considers the problem of extracting information from a random variable X that is relevant for prediction of a random variable Y. The relevance variable Y defines the high level task, which is a prediction task in this case. The extraction of relevant information from X is done by means of clustering of X into clusters $\tilde{X}$ [99]. In other words, IB is looking for the structure $\tilde{X}$ of X that is relevant for prediction of Y. The IB principle was further extended to graphical models in [92]. It is important to note that the requirement to compress X is an important part of the high level task in IB. In Chapter 5 we prove that if the high level task is solely the prediction of Y, clustering of X is not the optimal way to improve the predictions, but it is better to smooth the empirical conditional distributions.

The idea to look at generalization properties of clustering also appeared in [55, 56, 54], where Krupka and Tishby analyze a scenario where each

object instance has multiple properties and only a fraction of the properties is observed. The illustration they provide is the following: assume you are presented with multiple fruits and you observe their parameters such as size, color, and weight. Then you can cluster the fruits by their observed parameters in order to facilitate prediction of unobserved parameters such as taste and toxicity. The approach of Krupka and Tishby to this problem can be mapped onto the diagram in Figure 2.1.d. Namely, they have a well-defined objective to predict the unobserved properties, they define some loss function for incorrect predictions, they suggest an analysis of generalization properties of the clustering-based approach, and finally provide an algorithm to solve the corresponding optimization problem.

In section 2.2 we show that the framework suggested in Figure 2.1.d can also be used to analyze other forms of prediction based on clustering in addition to those suggested in [55, 56, 54]. Furthermore the bounding techniques developed in chapters 3 and 4 can be used to improve the bounds in [55, 56, 54].

2.1.1 The Minimum Description Length Principle (MDL)

A popular and related regularization approach that is applied in both supervised and unsupervised learning and deserves special attention is the Minimum Description Length Principle (MDL) [40]. The MDL principle suggests that a solution to a problem should be evaluated by the total description length of the train data, which is the description length of the selected model plus the description length of the data given the model. For example, one clustering solution is better than another if it describes more compactly the data at hand. The MDL approach has many of the properties that we want our solution to have: it enables comparison of different solutions in an objective manner, can perform model order selection and even compare solutions based on different optimization approaches (for example, PCA dimensionality reduction [33] and clustering). Thus, MDL provides a good indication that it is possible to answer these questions in an objective manner. However, MDL does not address the question of the expected

out-of-sample performance. In fact it has been shown that MDL solutions tend to overfit the data [49]. This observation is further supported by our experiments in Chapter 7. This is not very surprising, because MDL solves the question of data compression, but in situations when the obtained model is used for purposes other than compression of the train data, for example to make predictions on new data, MDL is not the appropriate criterion.

Interestingly, in the solutions we obtain in chapter 3, MDL solutions coincide with generalization bounds in situations when there is no noise in the data generating process, or, more precisely, when the selected model achieves zero empirical loss.

2.2 Case Study: Analysis of Data in the Form of a Matrix

In this section we show how to apply the workflow suggested in Figure 2.1.d to formulate the problem of analysis of data in the form of a matrix. "Data in the form of a matrix" is defined here as a data matrix in which rows and columns are of a "similar nature". For example, in a matrix of viewers by movies in collaborative filtering, the rows (viewers) are of a similar nature, as are the columns (movies). In other words, it makes sense to group together similar viewers, or to group together similar movies. However, a data matrix of cars by car parameters, e.g., fuel consumption, engine power, price, is not the kind of data considered in this section, since its columns are not of a similar nature - it does not make sense to group together engine power and car price.

We start with the observation discussed in [80, 83] that there are actually two types of data matrices that require different treatment. In the next two subsections we give an example of each of the problems and formulate the corresponding learning tasks and objective functionals. Then we show how these objective functionals are transformed into objectives for co-clustering. The co-clustering objective is further analyzed in chapter 4.

2.2.1 Functional Data and Objective Functional for Discriminative Prediction

Our first example is collaborative filtering [43]: here, one is given a matrix of viewers by movies with ratings, e.g. on a five-star scale, given by the viewers to the movies. The matrix is usually sparse, as most viewers have not seen all the movies. In this problem our task is usually to predict the missing entries. Thus, if we denote the viewer IDs by X_1, the movie IDs by X_2, and the rating values by Y, our goal is to build a discriminative predictor $q(Y|X_1, X_2)$. We use the term *functional data* to emphasize the fact that Y (the entries of the matrix) is a function of X_1 and X_2. Let $l(Y', Y)$ be a loss function for predicting Y', when the real rating is Y. The most commonly used loss functions are the zero-one loss $l(Y', Y) = 1 - \delta[Y', Y]$, where δ is the Kronecker delta function, absolute loss $l(Y', Y) = |Y' - Y|$, or quadratic loss $l(Y', Y) = (Y' - Y)^2$. Let $p(X_1, X_2, Y)$ be the true unknown distribution over $\langle X_1, X_2, Y \rangle$. A natural and commonly used way to define the learning objective for q is:

$$\min_{\text{Parameters of } q} \mathbb{E}_{p(X_1,X_2,Y)} \mathbb{E}_{q(Y'|X_1,X_2)} l(Y', Y). \tag{2.1}$$

In section 2.2.4 we show how this general learning objective is transformed into a learning objective for the co-clustering approach to solving this problem.

2.2.2 Co-occurrence Data and Objective Functional for Density Estimation

Our second example is the word-documents co-occurrence data analysis in text mining [93, 35, 32]. Word-documents co-occurrence matrices are matrices of words by documents with the number of times each word occurred in each document counted in the corresponding entries. If normalized, such a matrix can be regarded as an empirical joint probability distribution of words and documents; hence the name *co-occurrence data*. To illustrate the difference between co-occurrence data and functional data we point out

that if we extend the viewers-by-movies matrix in collaborative filtering by adding more viewers and more movies, the ratings already present will not change. However, if we extend the word-documents co-occurrence matrix by adding more words and more documents, the joint probability distribution (the entries in the normalized co-occurrence matrix) will change.

Although many researchers have analyzed this problem by clustering similar words and similar documents [93, 35, 32, 98], or by using probabilistic Latent Semantic Analysis (pLSA) and probabilistic Latent Semantic Indexing (pLSI) [45, 46] and other approaches, no clear learning task for this problem has been defined. In [83] we suggested one possible way to define such a task. Denote the words by X_1 and the documents by X_2. Assume that there is an unknown joint probability distribution $p(X_1, X_2)$ over words and documents. And assume that the co-occurrence matrix at hand is a sample from that distribution. Our task is to learn this joint probability distribution, or, in other words, to build a density estimator $q(X_1, X_2)$ that will provide good predictions of co-occurrence events generated by $p(X_1, X_2)$. We can write this objective formally as:

$$\min_{\text{Parameters of } q} -\mathbb{E}_{p(X_1,X_2)} \ln q(X_1, X_2). \tag{2.2}$$

The choice of the logarithmic loss is natural in the context of density estimation. In particular, it corresponds to the expected code length of an encoder that uses $q(X_1, X_2)$ to encode samples generated by $p(X_1, X_2)$ [27].

By evaluation of (2.2) it is possible to compare different approaches to solving this problem (e.g., co-clustering and pLSA), as well as perform model order selection. Next we show how the above objective is expressed in co-clustering and further analyze the co-clustering example in chapter 4.

2.2.3 Co-clustering

Co-clustering is a widely used method for analysis of data in the form of a matrix by simultaneous clustering of rows and columns of the matrix [9]. In this thesis we focus solely on co-clustering solutions that result in a grid form partition of the data matrix, as in Figure 2.2. This form of

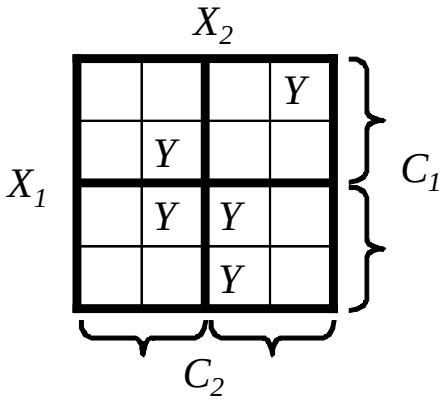


Figure 2.2: **Illustration of a hard grid form partition of a data matrix.** Soft grid partitions are considered as distributions over hard grid partitions.

co-clustering is also known as partitional co-clustering [9], checkerboard bi-clustering [23, 51], grid clustering [31, 82, 83], and box clustering. Note that some authors use the terms co-clustering and bi-clustering to refer to a simultaneous grouping of rows and columns that does not result in a grid-form partition of the whole data matrix [41, 64], but these forms of partitions are not discussed in this work. Note as well that we allow soft assignments of rows and columns to their clusters, as opposed to common co-clustering approaches which require hard assignments [32, 9]. Finally, the analysis presented here is not limited to two-dimensional data matrices.

In the past decade co-clustering has successfully been applied in multiple domains, including clustering of documents and words in text mining [93, 35, 32, 98], genes and experimental conditions in bioinformatics [23, 26, 51, 25], tokens and contexts in natural language processing [37, 73, 62], viewers and movies in recommender systems [38, 80], etc. Although Banerjee et. al. [9] suggested a unified framework for applying co-clustering in all the aforementioned domains, as we have already seen in sections 2.2.1 and 2.2.2 they are actually divided into two types that require different treatment. The objective functionals for discriminative prediction and for density estimation with co-clustering resulting from (2.1) and (2.2) are derived in the next two

sections.

2.2.4 Discriminative Prediction with Co-clustering

Let us return to the collaborative filtering example. One possible way to approach this problem is to cluster similar viewers, to cluster similar movies, and then to predict the missing entries by filling in the missing entries within each partition cell with the most frequent or average rating within that cell. More formally and generally, let us denote by $q(C_1|X_1)$ a stochastic rule that maps viewers to clusters of viewers, by $q(C_2|X_2)$ a stochastic rule that maps movies to clusters of movies, and by $q(Y|C_1, C_2)$ a stochastic rule that given a partition cell $\langle C_1, C_2 \rangle$ predicts a rating Y. Within this model:

$$q(Y|X_1, X_2) = \sum_{C_1, C_2} q(Y|C_1, C_2) q(C_1|X_1) q(C_2|X_2) \tag{2.3}$$

and the general objective in (2.1) is reduced to

$$\min_{\substack{q(C_1|X_1), q(C_2|X_2), \\ q(Y|C_1, C_2)}} \mathbb{E}_{p(X_1, X_2, Y)} \sum_{C_1, C_2, Y'} q(Y'|C_1, C_2) q(C_1|X_1) q(C_2|X_2) l(Y', Y), \tag{2.4}$$

where the set of distributions $\{q(Y|C_1, C_2), q(C_1|X_1), q(C_2|X_2)\}$ defines our co-clustering model.

In words, we say that we have found an interesting partition of viewers into clusters of viewers and of movies into clusters of movies if this partition is able to predict the ratings that the viewers assign to the movies. Similarly, in the context of gene expression data analysis we say that we have found an interesting partition of genes and conditions if this partition is able to predict the expression levels.

Bear in mind that we make no restrictions or assumptions regarding the true distribution $p(X_1, X_2, Y)$. Our only assumption is regarding the form of the prediction model $q(Y|X_1, X_2)$ we use, given in (2.3). In chapter 4 we provide a formal analysis of the objective (2.4) that was suggested in [82].

2.2.5 Density Estimation with Co-Clustering

Next we consider the word-documents co-occurrence data analysis example. Let $q(C_1|X_1)$ and $q(C_2|X_2)$ denote soft assignments of words to clusters of words and documents to clusters of documents respectively. Let $q(C_1, C_2)$ be our model for the joint probability distribution of word clusters and document clusters. We can then model the joint probability distribution of words and documents $q(X_1, X_2)$ as:

$$q(X_1, X_2) = \sum_{C_1, C_2} q(C_1, C_2) q(X_1|C_1) q(X_2|C_2). \tag{2.5}$$

Using Bayes' rule we can rewrite this as:

$$q(X_1, X_2) = \sum_{C_1, C_2} q(C_1, C_2) \frac{q(X_1)}{q(C_1)} q(C_1|X_1) \frac{q(X_2)}{q(C_2)} q(C_2|X_2), \tag{2.6}$$

which will be shown to be more convenient for analysis and application later in chapter 4. Then our objective (2.2) becomes:

$$\min_{\substack{q(C_1|X_1),\\ q(C_2|X_2)}} -\mathbb{E}_{p(X_1,X_2)} \ln \left[\sum_{C_1, C_2} q(C_1, C_2) \frac{q(X_1)}{q(C_1)} q(C_1|X_1) \frac{q(X_2)}{q(C_2)} q(C_2|X_2) \right], \tag{2.7}$$

where $\{q(C_1|X_1), q(C_2|X_2)\}$ define our co-clustering model (in chapter 4 we show that other distributions in (2.6) are induced by $\{q(C_1|X_1), q(C_2|X_2)\}$ and the empirical sample given in the data matrix, and thus are not part of the optimization).

In words, we have found a good clustering of words and a good clustering of documents if it is able to predict new co-occurrence events generated by $p(X_1, X_2)$. We stress that there are no assumptions or restrictions on $p(X_1, X_2)$ in (2.7). Our only assumptions are on the form of the prediction model we use, given in (2.6).

2.2.6 Other Approaches to Matrix Data Analysis and Discussion

Co-clustering is clearly not the only means to analyze data matrices. Other methods include pLSA and pLSI [45, 46] and other forms of matrix factorization [94]. To date we are only aware of generalization analysis of low-rank [95] and low-norm [96] outer product matrix approximations. The main contributions of our work in this respect can be summarized in the following points:

- In [82] we derived a generalization bound for discriminative prediction based on co-clustering (reproduced in Chapter 4). The bound enables theoretical comparison of co-clustering with other approaches to matrix data analysis.
- In [83] we suggested a possible definition of a high level task solved in co-occurrence data analysis (reproduced in section 2.2.2). This enables theoretical and practical comparison of co-clustering to other unrelated approaches to this problem, like LSA, that was not possible in the past.
- In [83] we suggest a generalization bound for co-occurrence data analysis based on co-clustering (reproduced in Chapter 4).

There is one appealing property that distinguishes co-clustering from other factor models like pLSA. This is the clear interpretation of the meaning of hidden variables of the model (cluster variables in this case). Whereas a cluster of viewers or a cluster of movies is easy to understand, the factors (hidden variables) returned by factor models like pLSA are much harder to interpret.

2.3 Beyond Matrix Data Analysis

The framework suggested in Figure 2.1.d can be applied to define generalization properties in other unsupervised learning models beyond co-clustering. One of the most popular unsupervised learning tasks is learning of mixtures

of Gaussians [33] and in general other mixture models. Following our framework we can state that we have learned a good mixture model if it is able to predict new events generated by the same unknown probability distribution that generated the train set. For example, let $p(\bar{X})$ be an unknown probability distribution over $\bar{X} \in \mathbb{R}^d$. Let $q(\bar{X}) = \sum_{i=1}^{k} q(C_i)q(\bar{X}|C_i)$ be the mixture model with k components (e.g., a mixture of k Gaussians). Our framework suggests analyzing and evaluating the mixture model by bounding $-\mathbb{E}_{p(\bar{X})} \ln q(\bar{X})$. This analysis is subject to future work.

In chapter 5 we show that models (2.3) and (2.6) suggested for co-clustering correspond to some simple forms of graphical models. We further suggest how the results of chapter 4 can be extended to analyze and derive generalization bounds for more general graphical models.

2.4 Beyond Prediction

The framework suggested in Figure 2.1.d does not limit the high level task to a prediction task. It is possible to analyze the advantages of structure-based models in other contexts. For example:

- Situations where we are not limited to a single prediction question, but rather a range of questions, when the exact questions we have to answer are not known in advance, for example, the touchstone formulation [88].

- Problems of control. For example, it may be easier to control or influence a process (e.g., our own hand or some tool) if we have a simple representation of its structure, rather than a high-dimensional kernel.

Structure-based models can also be preferable when computation or memory constraints are imposed. Moreover, it is not a-priori clear in questions of prediction whether structure-based approaches can outperform unstructured approaches like SVMs for example. Vapnik's well-known postulate states that "one should not solve a harder problem on the way to solving a simpler problem" [104]. Structure learning is an extra effort that is not

justified in the context of a pure prediction task. However, we know that as humans we perceive the world around us in a structured manner. Thus there must be advantages that we gain from the knowledge of structure. Otherwise, living beings would have developed some SVM classifiers during the process of evolution. The most distinctive advantage of structure-based models is *understanding* and *simplification* of the underlying processes and phenomenons. But in order to build computational models that are able to understand the data it is essential to quantify the notion of understanding or at least to be able to compare the level of understanding gained by different approaches, similar to the way we can measure who out of two students understood a course better. We conjecture that quantification of understanding should be done in the context of its potential applications. For example, one student can understand the course better to pass a written exam, but his schoolmate will outperform him in an oral exam, because it tests other type of understanding. Thus, the analysis of structure learning in the context of prediction tasks is just a small step on the way to quantification of knowledge and development of algorithms that are able to understand the data.

Part II

Analysis

Chapter 3

PAC-Bayesian Generalization Bounds

This chapter reviews some known tools and develops some novel ones for the analysis of learning algorithms. As mentioned in Chapter 2, one natural way to evaluate learning algorithms is to look at their expected performance on new data. A common approach to analysis of the expected performance is the Probably Approximately Correct (PAC) learning model [100, 50]. The most basic and distinctive property of this learning model is that the performance of a learning algorithm is evaluated with respect to the true distribution p that generates the data and no assumptions or restrictions on p are made.

The PAC learning model can be opposed to the Maximum Likelihood (ML) learning model [33]. In the ML learning model one is looking for a model q that maximizes the likelihood of the train set. In most real-life situations ML learning ends up overfitting the data.

A stronger learning model that the PAC model can be compared and contrasted with is the Bayesian learning model [63, 70]. The major difference between the PAC model and Bayesian inference is that in Bayesian inference it is assumed that the data are generated by some concept that belongs to the hypothesis class we are learning with. This assumption usually does not hold or holds only approximately when we analyze real-life data.

The Bayesian learning model is closely related to the MDL principle and MDL regularization [70, 40]. Similar to MDL, the Bayesian learning model does not provide guarantees on the expected performance on new data and thus is also prone to overfitting [49]. The PAC learning model makes no assumptions on the generating process and provides strict guarantees on the expected performance on new data. However, this often comes at the expense of underfitting the data [49]. This lack of tightness is the usual criticism made of PAC analysis.

The lack of tightness in PAC analysis is a result of its attempt to be as general as possible. In other words, posing no restrictions on the data generating distribution p requires the analysis to be strict enough to hold in the worst case. The fact that the real-life data in many situations do not follow the worst case scenario results in underfitting. However, this is only half of the reason for the lack of tightness of PAC analysis. The other half is the uniform treatment of all hypotheses in a hypothesis set. In other words, the PAC bounds hold uniformly for all the hypotheses in a given hypothesis set [104]. PAC-Bayesian bounds introduce non-uniform treatment of the hypotheses according to some prior partition of the hypothesis space [68, 76]. In situations where we are able to find some good partition of the hypothesis space we can improve the tightness of the bounds considerably. In the applications part we demonstrate that PAC-Bayesian bounds are able to achieve a remarkable 10%-20% distance from the test error.

Since the strength of PAC-Bayesian analysis lies in its ability to handle heterogeneous hypothesis spaces, this is also the domain where it has a significant advantage over traditional PAC analysis. There are two possible ways to introduce heterogeneity to a hypothesis space. The first is by using some "natural heterogeneity", as for example, in decision trees: we can partition the class of all possible decision trees, even of unlimited depth, into subclasses according to the tree depth. The second is by introducing a useful nonuniform partition of a homogeneous hypothesis space, for example, in the analysis of SVMs we can partition the class of all possible separating hyperplanes in $\mathbb{R}^d$ into subclasses according to the size of the margin [69, 59]. Note that the VC-dimension [102, 104, 31] of the class of decision

trees of unlimited depth as well as the separating hyperplanes in $\mathbb{R}^d$ for infinite-dimensional spaces (e.g., resulting from the use of RBF kernels [29]) is infinite. Thus, according to classical PAC theory it is impossible to learn with these hypothesis sets [50, 31, 104]. However, we will see in this chapter that PAC-Bayesian bounds enable learning with these hypothesis sets, although in a mildly different sense (thus we do not disprove the results of PAC theory, but only extend them slightly). Advantageously, the PAC-Bayesian bounds depend mainly on the complexity of the selected hypothesis and only slightly on the complexity of the whole hypothesis class, whereas PAC bounds characterize the whole hypothesis class by its VC-dimension and do not differentiate between hypotheses.

In structure learning the hypothesis class usually has a natural partition dictated by the complexity of the structure. Thus, PAC-Bayesian bounds are a handy tool for formal analysis of structure learning. In the next chapter we illustrate this using the example of co-clustering.

For completeness of the presentation we start by citing some well known results in measure concentration, then compare the PAC learning model to the learning model in PAC-Bayesian analysis and then gradually present PAC-Bayesian bounds and our results in this domain.

3.1 Background

3.1.1 Measure Concentration

The major work horse for a significant part of analysis done in computational learning theory is based on the effect of measure concentration. Measure concentration bounds suggest how fast (if at all) empirical observations converge to their expected values [19]. In this work two basic results of measure concentration are used:

Theorem 3.1 (Markov's inequality)**.** *Let X be any random variable and $\varepsilon > 0$, then*

$$P\{|X| \geq \varepsilon\} \leq \frac{\mathbb{E}|X|}{\varepsilon}. \tag{3.1}$$

Theorem 3.2 (Hoeffding's inequality [44]). *Let $X_1, .., X_N$ be independent bounded random variables such that X_i falls in the interval $[a_i, b_i]$ with probability one. Denote their average by $S_N = \frac{1}{N}\sum_{i=1}^{N} X_i$. Then for any $\varepsilon > 0$:*

$$P\{S_N - \mathbb{E}S_N \geq \varepsilon\} \leq e^{-2\varepsilon^2 N^2 / \sum_{i=1}^{N}(b_i - a_i)^2} \tag{3.2}$$

and

$$P\{S_N - \mathbb{E}S_N \leq -\varepsilon\} \leq e^{-2\varepsilon^2 N^2 / \sum_{i=1}^{N}(b_i - a_i)^2}. \tag{3.3}$$

An elegant proof of the latter theorem can be found in [19]. A related Chernoff-Hoeffding bound suggests a tighter concentration for zero-one variables, but before stating the bound we define the *Kullback-Leibler (KL) divergence* [57] between two probability distributions $P(X)$ and $Q(X)$ as:

$$D(P\|Q) = \mathbb{E}_{P(X)} \ln \frac{dP(X)}{dQ(X)}. \tag{3.4}$$

For two Bernoulli distributions $p(X)$ and $q(X)$ with biases p and q respectively we overload the definition as:

$$D_b(p\|q) = p \ln \frac{p}{q} + (1-p) \ln \frac{1-p}{1-q}. \tag{3.5}$$

The KL divergence between p and q can be bounded from below by their square distance [27]:

$$D_b(p\|q) \geq 2(p-q)^2. \tag{3.6}$$

With the above definitions we state the Chernoff-Hoeffding bound.

Theorem 3.3 (Chernoff-Hoeffding bound [24, 44]). *Let $X_1, .., X_N$ be independent bounded random variables such that $X_i \in \{0, 1\}$. Denote their average by $S_N = \frac{1}{N}\sum_{i=1}^{N} X_i$. Then for any $\varepsilon > 0$:*

$$P\{S_N - \mathbb{E}S_N \geq \varepsilon\} \leq e^{-D_b(\mathbb{E}S_N + \varepsilon || \mathbb{E}S_N)N} \leq e^{-2\varepsilon^2 N} \tag{3.7}$$

and

$$P\{S_N - \mathbb{E}S_N \leq -\varepsilon\} \leq e^{-D_b(\mathbb{E}S_N - \varepsilon || \mathbb{E}S_N)N} \leq e^{-2\varepsilon^2 N}, \tag{3.8}$$

In the context of learning theory it is often assumed that the train set is a set of N independent identically distributed (i.i.d.) samples. In such a situation the errors of a fixed predictor (hypothesis) h are also i.i.d. Let Z_i be the error of h on sample i and denote by $\hat{L}(h) = \frac{1}{N}\sum_{i=1}^{N} Z_i$ the empirical loss of h on the train set and by $L(h) = \mathbb{E}Z_i$ the expected loss of h. Assume we are in the context of zero-one loss, then by (3.8):

$$P\{L(h) \geq \hat{L}(h) + \varepsilon\} \leq e^{-2\varepsilon^2 N} \tag{3.9}$$

and by combination of (3.7) and (3.8):

$$P\{|L(h) - \hat{L}(h)| \geq \varepsilon\} \leq 2e^{-2\varepsilon^2 N}. \tag{3.10}$$

We can conclude that we can use the empirical loss of h, $\hat{L}(h)$, as an estimator of the expected loss of h, $L(h)$. The right-hand side of (3.9) or (3.10) is termed confidence and is denoted by δ, e.g. $\delta = 2e^{-2\varepsilon^2 N}$ in (3.10) which we read as: "with a probability greater than $1-\delta$, $\hat{L}(h)$ is ε-close to $L(h)$". This way of reading (3.10) is actually the reason for the name "PAC learning model": Probably - with a probability greater than $1-\delta$; Approximately - up to ε. The probability is over the choice of the sample (since each Z_i is a function of the sample), i.e., for most samples $|\hat{L}(h) - L(h)| < \varepsilon$, but there is a chance δ that for the sample we observe $|\hat{L}(h) - L(h)| \geq \varepsilon$.

3.1.2 Sample Complexity and Generalization Bounds

All learning algorithms do not operate with a single hypothesis, but rather with a class of hypotheses $\mathcal{H}$. Simultaneous analysis of what happens to $|\hat{L}(h) - L(h)|$ for all $h \in \mathcal{H}$ as a function of a random sample selection is one of the major technical challenges of computational learning theory. But before touching on this point it should be pointed out that there are three interrelated "players" in (3.9) or (3.10): precision ε, sample size N, and confidence δ. Thus, we can set two of them and see what happens to the remaining one. More specifically, we can set ε and inquire the minimal sample size N which is required to achieve ε-precision with a probability

greater than $1-\delta$. This is the *sample complexity* of a learning problem. Alternatively, we can set N and ask: "given a sample of size N how far can $L(h)$ be from $\hat{L}(h)$ with confidence greater than $1-\delta$". This is the question behind *generalization bounds*. As we will see below generalization bounds can be derived in situations where sample complexity bounds do not exist, but not vice-versa.

3.1.3 PAC vs. PAC-Bayes

PAC-Bayesian analysis exhibits the most basic feature of PAC learning - it makes no assumptions about the distribution that generates the data in the sense that PAC-Bayesian bounds hold for any distribution that could generate the data, even one that does not belong to the hypothesis class $\mathcal{H}$ we are learning with. However, there are multiple small details that distinguish PAC-Bayesian analysis from the classical PAC framework and eventually provide interesting results even in situations when a problem is not learnable in the strict PAC sense. We devote this section to a brief overview of PAC learning and its distinctions from PAC-Bayesian bounds.

The definition of PAC learnability states that [100, 50]:

Definition 3.1. *A hypothesis class $\mathcal{H}$ is PAC-learnable if for any distribution $\mathcal{D}$ over the data and for any $\varepsilon, \delta \in (0, \frac{1}{2})$ there exists N polynomial in ε, δ and a polynomial learning algorithm $\mathcal{A}$ that given an i.i.d. sample of size N from $\mathcal{D}$ returns $h \in \mathcal{H}$ that satisfies with a probability greater than $1-\delta$ over the sample selection and internal randomization of $\mathcal{A}$:*

$$L(h) \leq \inf_{h' \in \mathcal{H}} L(h') + \varepsilon. \tag{3.11}$$

This definition of PAC learnability refers to sample complexity. Moreover, (3.11) is a regret bound, as it looks at the distance between performance of the classifier returned by $\mathcal{A}$ and the performance of the best classifier in $\mathcal{H}$. Note that we do not know the value of $\inf_{h \in \mathcal{H}} L(h)$; thus the absolute value of $L(h)$ is known only implicitly through the fact that we know that it is close to $\hat{L}(h)$ with high probability. This point is the

first difference between PAC learning and PAC-Bayesian bounds. In PAC-Bayesian bounds we do not look at the distance between the performance of the classifier found and the performance of the best classifier in $\mathcal{H}$, but rather bound the expected performance $L(h)$ directly. We claim that in many practical situations it is sufficient to know the expected performance of the classifier returned. Moreover, for sufficiently rich hypothesis classes it may be impractical to get arbitrarily close to $\inf_{h\in\mathcal{H}} L(h)$ given a finite dataset, but still possible to find $h \in \mathcal{H}$ with satisfactory guarantees on $L(h)$ from a practical point of view. This makes it possible to obtain interesting results with PAC-Bayesian bounds even in situations when the hypothesis class $\mathcal{H}$ is not PAC learnable.

As already mentioned, for a hypothesis class $\mathcal{H}$ the statement (3.10) on $|\hat{L}(h) - L(h)|$ does not hold simultaneously for all $h \in \mathcal{H}$. If $\mathcal{H}$ is finite we can apply the union bound to obtain a statement of a form:

$$P\{\exists h \in \mathcal{H} : |\hat{L}(h) - L(h)| \geq \varepsilon\} \leq 2|\mathcal{H}|e^{-2\varepsilon^2 N} = 2e^{-2\varepsilon^2 N + \ln 2|\mathcal{H}|}, \quad (3.12)$$

where $|\mathcal{H}|$ is the cardinality of $\mathcal{H}$. Thus, for significantly large N [namely, for $N > \frac{1}{2\varepsilon^2}(\ln|\mathcal{H}| + \ln\frac{2}{\delta})$], $\hat{L}(h)$ will be sufficiently close to $L(h)$ for all $h \in \mathcal{H}$ and if we select

$$\hat{h} = \arg\min_{h\in\mathcal{H}} \hat{L}(h) \quad (3.13)$$

it will satisfy (3.11) with high probability [50]. Equation (3.13) is known as the *empirical risk minimization* approach [104]. It should be noted at this point that (3.13) treats all $h \in \mathcal{H}$ uniformly. The uniform treatment is inevitable if we want to obtain regret bound statements such as (3.11), since in order to do so we need $|\hat{L}(h) - L(h)|$ to be small for all $h \in \mathcal{H}$. However, if we bound $L(h)$ itself, but not its distance to the optimum, we can apply non-uniform treatment and derive much more interesting conclusions even for finite hypothesis classes, as in the example of co-clustering analyzed in Chapter 4.

The problem becomes more complicated when the hypothesis set $\mathcal{H}$ is infinite, since in that case we cannot apply the union bound, at least not

directly. In this situation there are two things we can do: cover the error space or cover the hypothesis space (or both).

The typical path of PAC analysis is to cover the error space. The first approach to covering the error space was developed by Vapnik and Chervonenkis [101, 102]. For a hypothesis space $\mathcal{H}$ of binary classification functions that map the inputs $X \in \mathcal{X}$ to zero-one labels $h : \mathcal{X} \mapsto \{0, 1\}$ Vapnik and Chervonenkis defined the VC-dimension of $\mathcal{H}$, which we denote by $VC(\mathcal{H})$, as the maximal number N of points in $\mathcal{X}$ that can be classified in all the 2^N possible ways by functions $h \in \mathcal{H}$ [102, 50]. For example, the VC-dimension of separating hyperplanes in $\mathbb{R}^d$ is $d + 1$, the VC-dimension of separating hyperplanes with margin γ for points bounded within a unit ball in $\mathbb{R}^d$ is bounded by $\min(\frac{1}{\gamma^2}, d) + 1$, and the VC-dimension of polynomials of degree d is $d + 1$. Vapnik and Chervonenkis further proved that for hypothesis classes with bounded VC-dimension:

$$P\{\exists h \in \mathcal{H} : |\hat{L}(h) - L(h)| \geq \varepsilon\} \leq 4 \left(\frac{2eN}{d}\right)^d e^{-N\varepsilon^2/8}, \tag{3.14}$$

where $d = VC(\mathcal{H})$ [102, 31, 20].

From (3.14) we can conclude that when N is large with respect to $\varepsilon^2, \ln d$, and $\ln \delta$ with high probability $|\hat{L}(h) - L(h)|$ will be small for all $h \in \mathcal{H}$ simultaneously and the empirical risk minimizer (3.13) will satisfy (3.11).

Bear in mind that (3.14) is ignorant of the sample and holds uniformly for all $h \in \mathcal{H}$. In later works Rademacher and Gaussian complexities were suggested to replace the VC-dimension in (3.14) [53, 10, 12, 20]. Rademacher and Gaussian complexities enable data-dependent estimation of the richness of a hypothesis class $\mathcal{H}$. They provide tighter bounds compared to VC-dimension bounds in situations when the data distribution p is concentrated on the regions of $\mathcal{X}$ in which the richness of $\mathcal{H}$ is not fully expressed. However, the treatment is still uniform for all $h \in \mathcal{H}$. Thus, although tighter than VC-bounds, they are still not satisfactory for nonhomogeneous hypothesis classes. Another technical problem with Rademacher and Gaussian complexities is that in most practical situations they cannot be written in a closed form and thus are problematic for gradient descent optimization.

Concentration of measure results such as (3.14) can also be used to derive generalization bounds. We can make the bounds even tighter by taking the one-sided version of (3.14):

$$P\{\exists h \in \mathcal{H} : L(h) \geq \hat{L}(h) + \varepsilon\} \leq 2 \left(\frac{2eN}{d}\right)^d e^{-N\varepsilon^2/8}. \tag{3.15}$$

By denoting the right-hand side of the inequality by δ we obtain that with a probability greater than $1 - \delta$:

$$L(h) < \hat{L}(h) + \sqrt{\frac{8\left(d \ln\left(\frac{2eN}{d}\right) + \ln\left(\frac{2}{\delta}\right)\right)}{N}} \tag{3.16}$$

for all $h \in \mathcal{H}$.

In the next section we present the first and the simplest approach to derivation of generalization bounds by covering the hypothesis space and then show how it can be combined with covering the error space.

3.2 Occam's Razor

To develop some intuition about an approach to hypothesis space covering we start with a simpler case, where $\mathcal{H}$ is countable.

Theorem 3.4 (Occam's razor)**.** *For any data generating distribution and for any "prior distribution" $\mathcal{P}(h)$ over $\mathcal{H}$ with a probability greater than $1-\delta$ over drawing an i.i.d. sample of size N, for all $h \in \mathcal{H}$ simultaneously:*

$$L(h) \leq \hat{L}(h) + \sqrt{\frac{-\ln \mathcal{P}(h) - \ln \delta}{2N}}. \tag{3.17}$$

Proof. The proof is fairly simple and provides a good illustration of what the "prior distribution" $\mathcal{P}(h)$ is. By Chernoff-Hoeffding's bound (3.8)

$$P\{L(h) - \hat{L}(h) \geq \varepsilon(h)\} \leq e^{-2N\varepsilon(h)^2} \tag{3.18}$$

for any given $h \in \mathcal{H}$. We use the notation $\varepsilon(h)$ to emphasize that the bound on $L(h) - \hat{L}(h)$ is designed for each hypothesis individually. We require that

$$e^{-2N\varepsilon(h)^2} \leq \mathcal{P}(h)\delta \tag{3.19}$$

for some prior $\mathcal{P}(h)$ that satisfies $\sum_{h\in\mathcal{H}} \mathcal{P}(h) \leq 1$. Then, by the union bound on (3.18)

$$L(h) \leq \hat{L}(h) + \varepsilon(h)$$

for all $h \in \mathcal{H}$ with a probability greater than $1 - \delta$. The minimal value of $\varepsilon(h)$ that satisfies the requirement (3.19) is

$$\varepsilon(h) = \sqrt{\frac{-\ln \mathcal{P}(h) - \ln \delta}{2N}},$$

which completes the proof. □

Note that the role of the prior $\mathcal{P}(h)$ is to define a different tradeoff between $\varepsilon(h)$ and $\delta(h)$ for each hypothesis $h \in \mathcal{H}$.

3.2.1 Application Example: Generalization Bound for Decision Trees

We illustrate the power and the beauty of Occam's razor bound by using a decision trees example. Let $\mathcal{T}$ be the hypothesis class of all binary decision trees of unlimited depth. For simplicity assume that all trees in $\mathcal{T}$ are complete (i.e., for a given $t \in \mathcal{T}$ all the leaves of t are at the same depth). Note that the VC-dimension of $\mathcal{T}$ is infinite, and thus it is not learnable in the strict PAC sense.

Let us denote by $\mathcal{T}_d$ the subset of $\mathcal{T}$ of trees of depth d. Then $\mathcal{T} = \bigcup_{d=0}^{\infty} \mathcal{T}_d$. For a given $t \in \mathcal{T}$ let us denote by $d(t)$ the depth of t. It is easy to verify that:

$$\mathcal{P}(t) = \frac{1}{2^{d(t)+1}} \frac{1}{2^{2^{d(t)}}} \tag{3.20}$$

is a legal prior over $\mathcal{T}$. By Occam's razor theorem, with a probability greater

than $1-\delta$:

$$L(t) < \hat{L}(t) + \sqrt{\frac{\ln(2)(2^{d(t)} + d(t) + 1) - \ln(\delta)}{2N}} \tag{3.21}$$

for all $t \in \mathcal{T}$.

Note that the VC-dimension of $\mathcal{T}_d$ is d. Thus, if we restrict ourselves to trees of depth d, by regular PAC bound (3.12) (or, more precisely, its one-sided version) we obtain that with a probability greater than $1-\delta$:

$$L(t) < \hat{L}(t) + \sqrt{\frac{\ln(2)(2^{d(t)}) - \ln(\delta)}{2N}} \tag{3.22}$$

for all $t \in \mathcal{T}_d$. By the negligible addition of $\ln(2)(d(t)+1)$ in (3.21) we are able to consider the whole $\mathcal{T}$ and for each $t \in \mathcal{T}$ the bound (3.21) is almost as tight as (3.22) which is obtained if we restrict the hypothesis space to $\mathcal{T}_d$.

In this example the prior $\mathcal{P}(t)$ defined in (3.20) is a "natural" prior over $\mathcal{T}$. The prior $\mathcal{P}(t)$ reflects the complexity of a tree t, which corresponds to the size of the subclass $\mathcal{T}_{d(t)}$ of "equivalently complex" trees that t belongs to. Or, looking at it the other way around, the prior $\mathcal{P}(t)$ defined in (3.20) suggests a meaningful partition of the infinite class $\mathcal{T}$ into subclasses $\mathcal{T}_d$ according to their complexity.

Prior knowledge about a problem can be used to increase the prior of certain trees and to improve the bound in the case where prior knowledge reflects reality. However, in this example it is useful only in situations where it can break the symmetry within $\mathcal{T}_d$-s. Since (3.21) is already very close to (3.22) prior knowledge about tree depth does not introduce a significant improvement in the bounds.

A slightly tighter (but also more complicated) analysis of the decision trees is suggested in [65].

3.2.2 Occam's Razor for Countable Unions of Bounded VC-dimension Hypothesis Classes

It is possible to combine Occam's razor technique for covering hypothesis spaces with techniques for covering the error spaces we briefly reviewed in section 3.1.3. This combination can be used to derive generalization bounds for uncountably large hypothesis spaces $\mathcal{H}$ in cases where $\mathcal{H}$ can be presented as a countable union of subspaces with a bounded VC-dimension. For example, $\mathcal{H}$ can be a hypothesis class of all polynomials of unbounded degree. Just as in the case of the decision trees, the VC-dimension of $\mathcal{H}$ is infinite.

We let $\mathcal{H} = \bigcup_{d=1}^{\infty} \mathcal{H}_d$, where $\mathcal{H}_d$ is a subspace of $\mathcal{H}$, which has VC-dimension d [for instance, a subset of polynomials of degree $(d-1)$]. We define a prior $\mathcal{P}(\mathcal{H}_d) = \frac{1}{2^d}$ over $\mathcal{H}$. Then we require that the right-hand side of (3.16) for each $\mathcal{H}_d$ be bounded by $\mathcal{P}(\mathcal{H}_d)\delta$. Following the lines of the proof of Occam's razor theorem for $\mathcal{P}(\mathcal{H}_d) = 2^{-d}$ we obtain that with a probability greater than $1-\delta$

$$L(h) < \hat{L}(h) + \sqrt{\frac{8\left(d(h)\ln\left(\frac{2eN}{d(h)}\right) + \ln\left(\frac{2}{\delta}\right) + d(h)\ln 2\right)}{N}} \tag{3.23}$$

for all $h \in \mathcal{H}$, where $d(h)$ is the VC-dimension of the subspace $\mathcal{H}_d$ that h belongs to. As in the example of decision trees, simultaneous treatment of all $\mathcal{H}_d$-s comes at a negligible price of $d\ln 2$.

The tradeoff between $\hat{L}(h)$ and $d(h)$ in (3.23) is closely related to the *structural risk minimization* principle suggested by Vapnik and Chervonenkis [103]. We find the Occam's razor approach to this derivation slightly more elegant. It also eliminates the requirement for nested $\mathcal{H}_d$-s.

3.2.3 Alternative Forms of Occam's Razor

The classical form of PAC-Bayesian bounds developed in the following sections is usually not in the explicit form of a bound on the distance between $L(h)$ and $\hat{L}(h)$, as in (3.17), but rather in an implicit, but tighter

form of a bound on $D_b(\hat{L}(h)\|L(h))$. For purposes of comparison between Occam's razor and the PAC-Bayesian bounds described later it should be noted that the Occam's razor bound can be written in the form of a bound on $D_b(\hat{L}(h)\|L(h))$ as well. If we use the tighter KL-divergence form of (3.8) and follow the lines of Occam's razor proof we obtain that with a probability greater than $1-\delta$

$$D_b(\hat{L}(h)\|L(h)) < \frac{-\ln \mathcal{P}(h) - \ln \delta}{N} \tag{3.24}$$

for all $h \in \mathcal{H}$ simultaneously. Furthermore, by the lower bound (3.6) on the KL-divergence we can recover (3.17) from (3.24).

We also point out that the requirement of the zero-one loss can be relaxed and the only requirement on $L(h)$ is to be bounded. The simplest way to obtain this relaxation is to use Hoeffding's inequality (3.3) instead of the Chernoff-Hoeffding bound in the proof of (3.17). By the convexity of the KL-divergence it is also possible to prove that if $L(h)$ is bounded within $[0,1]$ interval the KL-form of the Occam's razor bound (3.24) still holds [66]. And, of course, any bounded loss can be normalized to the $[0,1]$ interval.

3.2.4 Occam's Razor and the MDL Principle

This point is convenient to draw a relation between Occam's razor and the MDL principle. The MDL learning principle selects a hypothesis which has the minimal description length. The description length constitutes of the length of the hypothesis description plus the description of the data given the hypothesis. If we define a prior $\mathcal{P}(h)$ over $\mathcal{H}$ it is well known from the information theory that the number of bits required to describe h is $-\ln \mathcal{P}(h)$ [27]. The number of bits required to describe the data (the labels in this case) given h is $N\hat{L}(h)$. Thus, MDL selects $h \in \mathcal{H}$ which optimizes the tradeoff $\hat{L}(h) - \frac{1}{N}\ln \mathcal{P}(h)$. We note that the importance of $-\ln \mathcal{P}(h)$ decreases as $\frac{1}{N}$.

To compare the MDL principle with Occam's razor we bound the KL-divergence from below. For $q > p$ we have that $D_b(p\|q) \geq \frac{(q-p)^2}{2q}$. This inequality implies that if $D_b(p\|q) < x$ then $q \leq p + \sqrt{2px} + 2x$. By combining

this with (3.24) we obtain:

$$L(h) < \hat{L}(h) + \sqrt{\frac{2\hat{L}(h)(-\ln \mathcal{P}(h) - \ln \delta)}{N}} + \frac{2(-\ln \mathcal{P}(h) - \ln \delta)}{N}. \quad (3.25)$$

Thus, when $\hat{L}(h)$ is small the Occam's razor criterion is similar to the MDL criterion and the importance of $-\ln \mathcal{P}(h)$ decreases as $\frac{1}{N}$. However, if $\hat{L}(h)$ is large the importance of the prior decreases as $\frac{1}{\sqrt{N}}$. This result comes in line with other results on parameter estimation in statistics: when the noise level is low and the data generation process belongs to the hypothesis class, the speed of convergence of parameter estimations to their true values goes as $\frac{1}{N}$. However, if the noise level is high or the generation process does not belong to the hypothesis class, the speed of convergence goes as $\frac{1}{\sqrt{N}}$. Hence, in the latter case MDL is overfitting.

3.2.5 Randomized Predictors

Our further goal is to extend the technique of covering hypothesis spaces for general uncountable hypothesis spaces. This requires several modifications to Occam's razor one of which is to consider randomized predictors, which we define in this section. Let $\mathcal{Q}(h)$ be a distribution over a hypothesis space $\mathcal{H}$ (either countable or uncountable). A *randomized predictor* associated with $\mathcal{Q}$, and with a small abuse of notation denoted by $\mathcal{Q}$, is defined in the following way. For each sample x a hypothesis $h \in \mathcal{H}$ is drawn according to $\mathcal{Q}(h)$ and then used to make the prediction on x.

Note that in other works on PAC-Bayesian bounds $\mathcal{Q}$ is usually termed a *randomized classifier* [59]. In that context, $h(x)$ returns a label y of x as predicted by h. However, since this work extends the PAC-Bayesian framework beyond the classification scenario by using the same randomization technique we chose the term of randomized predictor. In this more general context $h(x)$ is a general function of x.

We further extend the definitions of the empirical and expected losses

for randomized predictors in the following way:

$$L(\mathcal{Q}) = \mathbb{E}_{\mathcal{Q}(h)} L(h) \tag{3.26}$$

and

$$\hat{L}(\mathcal{Q}) = \mathbb{E}_{\mathcal{Q}(h)} \hat{L}(h). \tag{3.27}$$

3.2.6 Occam's Razor for Randomized Predictors

In cases where the hypothesis space $\mathcal{H}$ is countable it is possible to apply Occam's razor to derive generalization bounds for randomized predictors. Since (3.24) holds for all $h \in \mathcal{H}$ simultaneously, for any distribution $\mathcal{Q}$ over $\mathcal{H}$ we have:

$$D_b(\hat{L}(\mathcal{Q}) \| L(\mathcal{Q})) < \frac{-\mathbb{E}_{\mathcal{Q}(h)} \ln \mathcal{P}(h) - \ln \delta}{N}. \tag{3.28}$$

Proof.

$$\begin{aligned} D_b(\hat{L}(\mathcal{Q}) \| L(\mathcal{Q})) &= D_b(\mathbb{E}_{\mathcal{Q}(h)} \hat{L}(h) \| \mathbb{E}_{\mathcal{Q}(h)} L(h)) \\ &\leq \mathbb{E}_{\mathcal{Q}(h)} D_b(\hat{L}(h) \| L(h)) \qquad (3.29) \\ &< \frac{-\mathbb{E}_{\mathcal{Q}(h)} \ln \mathcal{P}(h) - \ln \delta}{N}, \qquad (3.30) \end{aligned}$$

where (3.29) is by the convexity of the KL-divergence [27] and (3.30) is by the expectation $\mathbb{E}_{\mathcal{Q}(h)}$ of (3.24). □

3.3 PAC-Bayesian Generalization Bounds

PAC-Bayesian generalization bounds were suggested by McAllester [67, 68] as a tool to cover uncountably infinite hypothesis spaces. This comes at a price that the bounds hold for randomized classifiers, as defined in section 3.2.5, but cannot be directly applied to individual hypotheses, although there are some workarounds. Multiple successive works have suggested some improvements to the bound [7, 18, 66, 72] as well as some simplifications to its proof [76, 59, 66, 8]. In this work we chose to present the bound in the form suggested by Maurer [66], which is slightly tighter and simpler than

the original bound in [68]. In section 3.3.5 we mention some of the more sophisticated forms of the bound suggested in [7, 18, 72].

Theorem 3.5 (PAC-Bayesian bound for classification)**.** *For a hypothesis set $\mathcal{H}$, a prior distribution $\mathcal{P}$ over $\mathcal{H}$ and a loss function L bounded by 1, with a probability greater than $1-\delta$ over drawing a sample of size N, for all randomized classifiers $\mathcal{Q}$ simultaneously*:

$$D_b(\hat{L}(\mathcal{Q})\|L(\mathcal{Q})) \leq \frac{D(\mathcal{Q}\|\mathcal{P}) + \frac{1}{2}\ln(4N) - \ln\delta}{N}. \tag{3.31}$$

One of the most extensively studied applications of the PAC-Bayesian bound is the analysis of SVMs [68, 59, 4, 52, 72]. When a Gaussian prior over the linear separators is selected, the bounds provide a theoretical justification for the maximum margin principle in learning SVMs. They are also the tightest known bounds for SVMs. Derbeko et. al. [30] applied PAC-Bayesian bounds to analysis of transduction learning In [82] we suggested applying the PAC-Bayesian bound for the analysis of co-clustering which is presented in Chapter 4. Other applications include maximum margin analysis of structured classification [11]. A positive property that distinguishes PAC-Bayesian bounds is their explicit form of dependence on model parameters, which makes them easy to apply in optimization.

In [83] we suggested an extension of the PAC-Bayesian analysis technique and used it to derive a PAC-Bayesian bound for density estimation. Here we present a slightly better formulation of it.

Theorem 3.6. *Let $\mathcal{X}$ be the sample space and let $p(X)$ be an unknown and unrestricted distribution over $X \in \mathcal{X}$. Let $\mathcal{H}$ be a hypothesis class, such that each member h of $\mathcal{H}$ is a function $h : \mathcal{X} \mapsto \mathcal{Z}$, where $\mathcal{Z} = \{1, .., |Z|\}$. Let $p_h(Z) = P_{X\sim p(X)}\{h(X) = Z\}$ be the distribution over $\mathcal{Z}$ induced by $p(X)$ and h. Let $\mathcal{P}$ be a prior distribution over $\mathcal{H}$. Let $\mathcal{Q}$ be an arbitrary distribution over $\mathcal{H}$ and $p_{\mathcal{Q}}(Z) = \mathbb{E}_{\mathcal{Q}(h)}p_h(Z)$ a distribution over Z induced by $p(X)$ and $\mathcal{Q}$. Let S be an i.i.d. sample of size N generated according to $p(X)$ and let $\hat{p}(X)$ be the empirical distribution over $\mathcal{X}$ corresponding to S. Let $\hat{p}_h(Z) = P_{X\sim\hat{p}(X)}\{h(X) = Z\}$ be the empirical distribution over Z*

corresponding to h and S and $\hat{p}_{\mathcal{Q}}(Z) = \mathbb{E}_{\mathcal{Q}(h)}\hat{p}_h(Z)$. Then with a probability greater than $1-\delta$ for all possible $\mathcal{Q}$ simultaneously:

$$D(\hat{p}_{\mathcal{Q}}(Z)\|p_{\mathcal{Q}}(Z)) \leq \frac{D(\mathcal{Q}\|\mathcal{P}) + (|Z|-1)\ln(N+1) - \ln\delta}{N}. \tag{3.32}$$

We have further demonstrated that the PAC-Bayesian bound for classification (3.31) is a special case of the PAC-Bayesian bound for density estimation, when we consider Z as the error variable. The proof of theorem 3.6 is surprisingly simple and reveals a close relation between the PAC-Bayesian theorems and the method of types in information theory [27]. Further relations between the PAC-Bayesian bounds, information theory and statistical mechanics are discussed in [21].

The proof of theorem 3.6 is based on three simple steps. First we bound the expectation of the exponent of the divergence, $\mathbb{E}e^{ND(\hat{p}_h(X)\|p_h(X))}$ for a single hypothesis h. Then we bound the same exponent of the divergence, when h is selected at random according to $\mathcal{P}(h)$ and apply Markov's inequality to bound the probability that $\mathbb{E}_{\mathcal{P}(h)}e^{ND(\hat{p}_h(Z)\|p_h(Z))}$ diverges significantly from its expectation. Finally, we apply a change of measure inequality to infer (3.32) for all $\mathcal{Q}$. The details of the proof are presented in the following sections. Some further discussion is presented afterwards.

3.3.1 The Law of Large Numbers

We first analyze the rate of convergence of empirical distributions over finite domains around their true values. The following result is based on the method of types in information theory [27].

Theorem 3.7. *Let $S = \{X_1, .., X_N\}$ be i.i.d. distributed by $p(X)$ and let $|X|$ be the cardinality of X. Denote by $\hat{p}(X)$ the empirical distribution of S. Then*:

$$\mathbb{E}_S e^{ND(\hat{p}(X)\|p(X))} \leq (N+1)^{|X|-1}. \tag{3.33}$$

Proof. Enumerate the possible values of X by $1, .., |X|$ and let n_i count the number of occurrences of value i. Let p_i denote the probability of value i and $\hat{p}_i = \frac{n_i}{N}$ be its empirical counterpart. Let $H(\hat{p}) = -\sum_i \hat{p}_i \ln \hat{p}_i$ be the

empirical entropy. Then:

$$\begin{aligned}\mathbb{E}_S e^{ND(\hat{p}(X)\|p(X))} &= \sum_{\substack{n_1,..,n_{|X|}:\\ \sum_i n_i=N}} \binom{N}{n_1,..,n_{|X|}} \cdot \prod_{i=1}^{|X|} p_i^{N\hat{p}_i} \cdot e^{ND(\hat{p}(X)\|p(X))} \\ &\leq \sum_{\substack{n_1,..,n_{|X|}:\\ \sum_i n_i=N}} e^{NH(\hat{p})} \cdot e^{N\sum_i \hat{p}_i \ln p_i} \cdot e^{ND(\hat{p}(X)\|p(X))} \qquad (3.34) \\ &= \sum_{\substack{n_1,..,n_{|X|}:\\ \sum_i n_i=N}} 1 = \binom{N+|X|-1}{|X|-1} \leq (N+1)^{|X|-1}. \qquad (3.35)\end{aligned}$$

In (3.34) we use the $\binom{N}{n_1,..,n_{|X|}} \leq e^{NH(\hat{p})}$ bound on the multinomial coefficient, which counts the number of sequences with a fixed cardinality profile (type) $n_1,..,n_{|X|}$ [27]. In the second equality in (3.35) the number of ways to choose n_i-s equals the number of ways we can place $|X|-1$ ones in a sequence of $N+|X|-1$ ones and zeros, where ones symbolize a partition of zeros ("balls") into $|X|$ bins. □

Some Corollaries of Theorem 3.7

Note in passing that it is straightforward to recover theorem 12.2.1 in [27] from theorem 3.7. We even suggest a small improvement over it:

Theorem 3.8 (12.2.1 in Cover and Thomas, 1991)**.** *Under the notations of theorem 3.7*:

$$P\left\{D(\hat{p}(X)\|p(X)) \geq \varepsilon\right\} \leq e^{-N\varepsilon+(|X|-1)\ln(N+1)}, \qquad (3.36)$$

or, equivalently, with a probability greater than $1-\delta$:

$$D(\hat{p}(X)\|p(X)) \leq \frac{(|X|-1)\ln(N+1)-\ln\delta}{N}. \qquad (3.37)$$

Proof. By Markov's inequality and theorem 3.7:

$$\begin{aligned} P\{D(\hat{p}(X)\|p(X)) \geq \varepsilon\} &= P\{e^{ND(\hat{p}(X)\|p(X))} \geq e^{N\varepsilon}\} \\ &\leq \frac{\mathbb{E}e^{ND(\hat{p}(X)\|p(X))}}{e^{N\varepsilon}} \\ &\leq \frac{(N+1)^{|X|-1}}{e^{N\varepsilon}} = e^{-N\varepsilon+(|X|-1)\ln(N+1)}. \end{aligned}$$

□

3.3.2 Change of Measure Inequality

Simultaneous treatment of all possible distributions (measures) $\mathcal{Q}$ over $\mathcal{H}$ is done by relating them all to a single reference (prior) measure $\mathcal{P}$. We call this relation a *change of measure inequality*. It appears in the proof of the PAC-Bayesian theorem in [69] and was formulated as a standalone result in [8]. Banerjee [8] terms it a *compression lemma*, however we find the name "change of measure inequality" more appropriate to its nature and usage. The inequality is a simple consequence of Jensen's inequality.

Lemma 3.1 (Change of Measure Inequality)**.** *For any measurable function $\phi(h)$ on $\mathcal{H}$ and any distributions $\mathcal{P}$ and $\mathcal{Q}$ on $\mathcal{H}$, we have:*

$$\mathbb{E}_{\mathcal{Q}(h)}\phi(h) \leq D(\mathcal{Q}\|\mathcal{P}) + \ln \mathbb{E}_{\mathcal{P}(h)}e^{\phi(h)}. \tag{3.38}$$

Proof. For any measurable function $\phi(h)$, we have:

$$\begin{aligned} \mathbb{E}_{\mathcal{Q}(h)}\phi(h) &= \mathbb{E}_{\mathcal{Q}(h)} \ln \left(\frac{d\mathcal{Q}(h)}{d\mathcal{P}(h)} \cdot e^{\phi(h)} \cdot \frac{d\mathcal{P}(h)}{d\mathcal{Q}(h)} \right) \\ &= D(\mathcal{Q}\|\mathcal{P}) + \mathbb{E}_{\mathcal{Q}(h)} \ln \left(e^{\phi(h)} \cdot \frac{d\mathcal{P}(h)}{d\mathcal{Q}(h)} \right) \\ &\leq D(\mathcal{Q}\|\mathcal{P}) + \ln \mathbb{E}_{\mathcal{Q}(h)} \left(e^{\phi(h)} \cdot \frac{d\mathcal{P}(h)}{d\mathcal{Q}(h)} \right) \qquad (3.39) \\ &= D(\mathcal{Q}\|\mathcal{P}) + \ln \mathbb{E}_{\mathcal{P}(h)}e^{\phi(h)}, \qquad (3.40) \end{aligned}$$

where (3.39) is by Jensen's inequality. □

3.3.3 Proof of the PAC-Bayesian Generalization Bound for Density Estimation

We apply the results of the previous two sections to prove the PAC-Bayesian generalization bound for density estimation in theorem 3.6.

Proof (Theorem 3.6). Let $S = \{X_1, .., X_N\}$ be an i.i.d. sample according to $p(X)$ and let $\{Z_1^h, .., Z_N^h\} = \{h(X_1), .., h(X_N)\}$. Then Z_i^h are i.i.d. distributed according to $p_h(Z)$ and we denote their empirical distribution by $\hat{p}_h(Z)$. Let $\phi(h, S, p) = ND(\hat{p}_h(Z)\|p_h(Z))$. Then:

$$\begin{aligned} ND(\hat{p}_{\mathcal{Q}}(Z)\|p_{\mathcal{Q}}(Z)) &= ND(\mathbb{E}_{\mathcal{Q}(h)}\hat{p}_h(Z)\|\mathbb{E}_{\mathcal{Q}(h)}p_h(Z)) \\ &\leq \mathbb{E}_{\mathcal{Q}(h)}ND(\hat{p}_h(Z)\|p_h(Z)) \qquad (3.41) \\ &\leq D(\mathcal{Q}\|\mathcal{P}) + \ln \mathbb{E}_{\mathcal{P}(h)}e^{ND(\hat{p}_h(Z)\|p_h(Z))}, \qquad (3.42) \end{aligned}$$

where (3.41) is by the convexity of the KL-divergence [27] and (3.42) is by the change of measure inequality. To obtain (3.32) it is left to bound $\mathbb{E}_{\mathcal{P}(h)}e^{ND(\hat{p}_h(Z)\|p_h(Z))}$:

$$\mathbb{E}_S\left[\mathbb{E}_{\mathcal{P}(h)}e^{ND(\hat{p}_h(Z)\|p_h(Z))}\right] = \mathbb{E}_{\mathcal{P}(h)}\left[\mathbb{E}_S e^{ND(\hat{p}_h(Z)\|p_h(Z))}\right] \leq (N+1)^{|Z|-1}, \qquad (3.43)$$

where the last inequality is justified by the fact that $\mathbb{E}_S e^{ND(\hat{p}_h(Z)\|p_h(Z))} \leq (N+1)^{|Z|-1}$ for each h individually according to (3.33). By (3.43) and Markov's inequality we conclude that with a probability of at least $1-\delta$ over S:

$$\mathbb{E}_{\mathcal{P}(h)}e^{ND(\hat{p}_h(Z)\|p_h(Z))} \leq \frac{(N+1)^{|Z|-1}}{\delta}. \qquad (3.44)$$

Substituting this into (3.42) and normalizing by N provides (3.32). □

3.3.4 Proof of the PAC-Bayesian Bound for Classification

To recover the PAC-Bayesian theorem 3.5 from theorem 3.6 in the case of zero-one loss let Z be the zero-one error variable. In this case h maps a

sample $\langle x, y\rangle$ to $Z = \mathbb{I}_{h(x)=y}$. Then $L(h) = \mathbb{E}Z = p_h\{Z = 1\}$ and $L(\mathcal{Q}) = p_{\mathcal{Q}}\{Z = 1\}$. As well, $|Z| = 2$. Substituting this into (3.32) we obtain (3.31) up to a factor of $\frac{1}{2}\ln N$. By the convexity of $D(\hat{L}(\mathcal{Q})\|L(\mathcal{Q}))$ it is possible to show that the result holds for any loss bounded by 1 [66]. The improvement of $\frac{1}{2}\ln N$ in theorem 3.5 is achieved by showing that in the case where $L(h) = \mathbb{E}Z$ the expectation $\mathbb{E}e^{ND(\hat{L}(h)\|L(h))} \leq 2\sqrt{N}$ instead of the more general bound $\mathbb{E}e^{ND(\hat{p}(Z)\|p(Z))} \leq (N+1)^{|Z|-1}$ for distributions we have in theorem 3.7.

3.3.5 Remarks

Finite hypothesis sets both (3.28) and (3.31) hold. By writing $D(\mathcal{Q}\|\mathcal{P}) = -H(\mathcal{Q}) - \mathbb{E}_{\mathcal{Q}(h)}\ln\mathcal{P}(h)$ we find that (3.31) is tighter than (3.28) when $H(\mathcal{Q}) > \frac{1}{2}\ln(4N)$. However, as N increases $H(\mathcal{Q})$ is likely to decrease and in the limit of large N (3.28) is tighter. Some authors suggest slightly different forms of the PAC-Bayesian bound, where the $\frac{1}{2}\ln(4N)$ term is either reduced or completely eliminated at the cost of increasing the $D(\mathcal{Q}\|\mathcal{P})$ coefficient [7, 18]. In [21, 72] instead of bounding $D_b(\hat{L}(\mathcal{Q})\|L(\mathcal{Q}))$ bounding of functions of the form $\mathcal{F}(L(\mathcal{Q})) - C\hat{L}(\mathcal{Q})$ for constant C and convex $\mathcal{F}$ is considered, which can also provide slightly tighter results if optimized with respect to C.

The tradeoff between $\hat{L}(\mathcal{Q})$ and $D(\mathcal{Q}\|\mathcal{P})$ in the PAC-Bayesian bounds has a tight relation to the maximum entropy principle in learning and statistical mechanics [48, 34, 21, 89]. This point is further discussed in [21, 89].

3.3.6 Smoothing

In some applications, and in particular in the co-clustering analysis we discuss in the next chapter, it may be of interest to find an estimate $q(Z)$ of $p_{\mathcal{Q}}(Z)$. It is also natural to evaluate $q(Z)$ by its logarithmic loss $-\mathbb{E}_{p_{\mathcal{Q}}(Z)}\ln q(Z)$. The latter corresponds to the expected code length of encoder q when samples are generated by $p_{\mathcal{Q}}$ [27]. Although we bounded $D(\hat{p}_{\mathcal{Q}}(Z)\|p_{\mathcal{Q}}(Z))$ in theorem 3.6, $\hat{p}_{\mathcal{Q}}(Z)$ cannot be used as an estimator for $p_{\mathcal{Q}}(Z)$ since it is not bounded from zero. To cope with this, we define a smoothed version of $\hat{p}$

we call q:

$$q_h(Z) = \frac{\hat{p}_h(Z) + \gamma}{1 + \gamma|Z|}, \tag{3.45}$$

$$q_{\mathcal{Q}}(Z) = \mathbb{E}_{\mathcal{Q}(h)} q_h(Z) = \frac{\hat{p}_{\mathcal{Q}}(Z) + \gamma}{1 + \gamma|Z|}. \tag{3.46}$$

In the following theorem we show that if $D(\hat{p}_{\mathcal{Q}}(Z)\|p_{\mathcal{Q}}(Z)) \leq \varepsilon(\mathcal{Q})$ and $\gamma = \frac{\sqrt{\varepsilon(\mathcal{Q})/2}}{|Z|}$, then $-\mathbb{E}_{p_{\mathcal{Q}}(Z)} \ln q_{\mathcal{Q}}(Z)$ is roughly within $\pm\sqrt{\varepsilon(\mathcal{Q})/2} \ln |Z|$ range around $H(\hat{p}_{\mathcal{Q}}(Z))$. The bound on $D(\hat{p}_{\mathcal{Q}}(Z)\|p_{\mathcal{Q}}(Z))$ is naturally obtained by theorem 3.6. Thus, the performance of the density estimator $q_{\mathcal{Q}}$ is optimized by distribution $\mathcal{Q}$ that minimizes the tradeoff between $H(\hat{p}_{\mathcal{Q}}(Z))$ and $\frac{1}{N} D(\mathcal{Q}\|\mathcal{P})$.

Note that for a uniform distribution $u(Z) = \frac{1}{|Z|}$ the value of $-\mathbb{E}_{p(Z)} \ln u(Z) = \ln |Z|$. Thus, the theorem is interesting when $\sqrt{\varepsilon(\mathcal{Q})/2}$ is significantly smaller than 1. For technical reasons we encounter in the proofs of the next chapter, the upper bound in the following theorem is stated for $-\mathbb{E}_{p_{\mathcal{Q}}(Z)} \ln q_{\mathcal{Q}}(Z)$ and for $-\mathbb{E}_{\mathcal{Q}(h)} \mathbb{E}_{p_h(Z)} \ln q_h(Z)$. We also denote $\varepsilon = \varepsilon(\mathcal{Q})$ for brevity.

Theorem 3.9. *Let Z be a random variable distributed according to $p_{\mathcal{Q}}(Z)$ and assume that $D(\hat{p}_{\mathcal{Q}}(Z)\|p_{\mathcal{Q}}(Z)) \leq \varepsilon$. Then $-\mathbb{E}_{p_{\mathcal{Q}}(Z)} \ln q_{\mathcal{Q}}(Z)$ is minimized by $\gamma = \frac{\sqrt{\varepsilon/2}}{|Z|}$. For this value of γ the following inequalities hold:*

$$-\mathbb{E}_{\mathcal{Q}(h)} \mathbb{E}_{p_h(Z)} \ln q_h(Z) \leq H(\hat{p}_{\mathcal{Q}}(Z)) + \sqrt{\varepsilon/2} \ln |Z| + \phi(\varepsilon), \tag{3.47}$$

$$-\mathbb{E}_{p_{\mathcal{Q}}(Z)} \ln q_{\mathcal{Q}}(Z) \leq H(\hat{p}_{\mathcal{Q}}(Z)) + \sqrt{\varepsilon/2} \ln |Z| + \phi(\varepsilon), \tag{3.48}$$

$$-\mathbb{E}_{p_{\mathcal{Q}}(Z)} \ln q_{\mathcal{Q}}(Z) \geq H(\hat{p}_{\mathcal{Q}}(Z)) - \sqrt{\varepsilon/2} \ln |Z| - \psi(\varepsilon), \tag{3.49}$$

where:

$$\psi(\varepsilon) = \sqrt{\frac{\varepsilon}{2}} \ln \frac{1 + \sqrt{\frac{\varepsilon}{2}}}{\sqrt{\frac{\varepsilon}{2}}} \quad \textit{and} \quad \phi(\varepsilon) = \psi(\varepsilon) + \ln(1 + \sqrt{\frac{\varepsilon}{2}}).$$

Note that both $\phi(\varepsilon)$ and $\psi(\varepsilon)$ go to zero approximately as $-\sqrt{\varepsilon/2} \ln \sqrt{\varepsilon/2}$.

Proof. Recall that by the KL-divergence bound on the L_1 norm [27]:

$$\|\hat{p}_{\mathcal{Q}}(Z) - p_{\mathcal{Q}}(Z)\|_1 \leq \sqrt{2D(\hat{p}_{\mathcal{Q}}(Z)\|p_{\mathcal{Q}}(Z))} \leq \sqrt{2\varepsilon}. \tag{3.50}$$

For the proof of (3.47):

$$\begin{aligned}
-\mathbb{E}_{\mathcal{Q}(h)}\mathbb{E}_{p_h(Z)} \ln q_h(Z) &= \mathbb{E}_{\mathcal{Q}(h)}\mathbb{E}_{[\hat{p}_h(Z)-p_h(Z)]} \ln q_h(Z) - \mathbb{E}_{\mathcal{Q}(h)}\mathbb{E}_{\hat{p}_h(Z)} \ln q_h(Z) \\
&= \mathbb{E}_{\mathcal{Q}(h)}\mathbb{E}_{[\hat{p}_h(Z)-p_h(Z)]} \ln \frac{\hat{p}_h(Z)+\gamma}{1+\gamma|Z|} - \mathbb{E}_{\mathcal{Q}(h)}\mathbb{E}_{\hat{p}_h(Z)} \ln \frac{\hat{p}_h(Z)+\gamma}{1+\gamma|Z|} \\
&\leq -\frac{1}{2}\|\hat{p}_{\mathcal{Q}}(Z) - p_{\mathcal{Q}}(Z)\|_1 \ln \frac{\gamma}{1+\gamma|Z|} + \mathbb{E}_{\mathcal{Q}(h)} H(\hat{p}_h(Z)) + \ln(1+\gamma|Z|) \\
&\leq H(\hat{p}_{\mathcal{Q}}(Z)) - \sqrt{\varepsilon/2} \ln \frac{\gamma}{1+\gamma|Z|} + \ln(1+\gamma|Z|), \qquad (3.51)
\end{aligned}$$

where (3.51) is justified by the concavity of the entropy function H and (3.50). By differentiation (3.51) is minimized by $\gamma = \frac{\sqrt{\varepsilon/2}}{|Z|}$. By substitution of this value of γ into (3.51) we obtain (3.47). Inequality (3.48) is justified by (3.47) and the concavity of the ln function. Finally, we prove the lower bound (3.49):

$$\begin{aligned}
-\mathbb{E}_{p_{\mathcal{Q}}(Z)} \ln q_{\mathcal{Q}}(Z) &= \mathbb{E}_{[\hat{p}_{\mathcal{Q}}(Z)-p_{\mathcal{Q}}(Z)]} \ln q_{\mathcal{Q}}(Z) - \mathbb{E}_{\hat{p}_{\mathcal{Q}}(Z)} \ln q_{\mathcal{Q}}(Z) \\
&\geq -\frac{1}{2}\|\hat{p}_{\mathcal{Q}}(Z) - p_{\mathcal{Q}}(Z)\|_1 \ln \frac{1+\gamma|Z|}{\gamma} + H(\hat{p}_{\mathcal{Q}}(Z)) \\
&\geq H(\hat{p}_{\mathcal{Q}}(Z)) - \sqrt{\varepsilon/2} \ln \frac{|Z|(1+\sqrt{\varepsilon/2})}{\sqrt{\varepsilon/2}}.
\end{aligned}$$

□

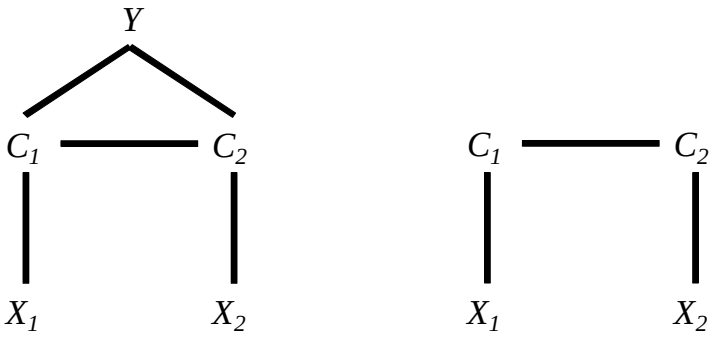
Chapter 4

PAC-Bayesian Analysis of Co-clustering

In sections 2.2.3, 2.2.4, and 2.2.5 we presented the co-clustering approach to discriminative prediction and density estimation. Here we apply the PAC-Bayesian generalization bounds developed in the previous chapter to analyze the co-clustering approach. We begin with the co-clustering approach to discriminative prediction, which is slightly easier in terms of presentation. Below it is presented in a more general and formal way than in the preliminary definition in Chapter 2.

4.1 PAC-Bayesian Analysis of Discriminative Prediction with Grid Clustering

Let $\mathcal{X}_1 \times .. \times \mathcal{X}_d \times \mathcal{Y}$ be a $(d+1)$-dimensional product space. We assume that each $\mathcal{X}_i$ is categorical and its cardinality is fixed and known and we denote it by $|\mathcal{X}_i| = n_i$. We also assume that $\mathcal{Y}$ is finite with cardinality $|Y|$ and that a bounded loss function $l(Y, Y')$ for predicting Y' instead of Y is given. As an example consider collaborative filtering. In collaborative filtering $d = 2$, $\mathcal{X}_1$ is the space of viewers, n_1 is the number of viewers, $\mathcal{X}_2$ is the space of movies, n_2 is the number of movies, and $\mathcal{Y}$ is the space of the



(a) Graphical Model for Discriminative Prediction

(b) Graphical Model for Density Estimation

Figure 4.1: **Illustration of graphical models corresponding to discriminative prediction** (4.1) **and density estimation** (4.15) **models.** The illustrations are for the case of $d = 2$.

ratings (e.g., on a five-star scale). $l(Y, Y')$ can be, for example, an absolute loss $l(Y, Y') = |Y - Y'|$ or a quadratic loss $l(Y, Y') = (Y - Y')^2$. There is no natural metric on the space of viewers and on the space of movies; thus both $\mathcal{X}_1$ and $\mathcal{X}_2$ are categorical.

We assume there exists an unknown probability distribution $p(X_1, .., X_d, Y)$ over the product space. We further assume that we are given an i.i.d. sample of size N generated according to $p(X_1, .., X_d, Y)$. We use $\hat{p}(X_1, .., X_d, Y)$ to denote the empirical frequencies of $(d + 1)$-tuples $\langle X_1, .., X_d, Y \rangle$ in the sample. We consider the following form of discriminative predictors (which is a more general form of (2.3)):

$$q(Y|X_1, .., X_d) = \sum_{C_1,..,C_d} q(Y|C_1, .., C_d) \prod_{i=1}^{d} q(C_i|X_i). \tag{4.1}$$

The hidden variables $C_1, .., C_d$ represent a clustering of the observed variables $X_1, .., X_d$. The hidden variable C_i accepts values in $\{1, .., m_i\}$, where $m_i = |C_i|$ denotes the number of clusters used along dimension i. The conditional probability distribution $q(C_i|X_i)$ represents the probability of mapping (assigning) X_i to cluster C_i. The conditional probability $q(Y|C_1, .., C_d)$

represents the probability of assigning label Y to cell $\langle C_1, .., C_d \rangle$ in the cluster product space. The prediction model (4.1) corresponds to the graphical model in Figure 4.1.a. The free parameters of the model are the conditional distributions $\{q(C_i|X_i)\}_{i=1}^d$ and $q(Y|C_1, .., C_d)$. We denote these collectively by $\mathcal{Q} = \left\{\{q(C_i|X_i)\}_{i=1}^d, q(Y|C_1, .., C_d)\right\}$. In the next section we show that (4.1) corresponds to a randomized prediction strategy. We further denote:

$$L(\mathcal{Q}) = \mathbb{E}_{p(X_1,..,X_d,Y)}\mathbb{E}_{q(Y|X_1,..,X_d)}l(Y, Y') \tag{4.2}$$

and

$$\hat{L}(\mathcal{Q}) = \mathbb{E}_{\hat{p}(X_1,..,X_d,Y)}\mathbb{E}_{q(Y|X_1,..,X_d)}l(Y, Y'), \tag{4.3}$$

where $q(Y|X_1, .., X_d)$ is defined by (4.1).

We define

$$\tilde{I}(X_i; C_i) = \frac{1}{n_i}\sum_{x_i,c_i} q(c_i|x_i) \ln \frac{q(c_i|x_i)}{\tilde{q}(c_i)}, \tag{4.4}$$

where $x_i \in \mathcal{X}_i$ are the possible values of X_i, c_i are the possible values of C_i, and

$$\tilde{q}(c_i) = \frac{1}{n_i}\sum_{x_i} q(c_i|x_i). \tag{4.5}$$

$\tilde{q}(c_i)$ is the marginal distribution over C_i corresponding to $q(C_i|X_i)$ and a *uniform* distribution $q_u(x_i) = \frac{1}{n_i}$ over X_i and $\tilde{I}(X_i; C_i)$ is the mutual information corresponding to the joint distribution $q(x_i, c_i) = \frac{1}{n_i}q(c_i|x_i)$ defined by $q(c_i|x_i)$ and the uniform distribution over X_i.

With the above definitions we can state the following theorem.

Theorem 4.1. *For any probability measure $p(X_1, .., X_d, Y)$ over $\mathcal{X}_1 \times .. \times \mathcal{X}_d \times \mathcal{Y}$ and for any loss function l bounded by 1, with a probability of at least $1 - \delta$ over a selection of an i.i.d. sample S of size N according to p, for all randomized classifiers $\mathcal{Q} = \left\{\{q(C_i|X_i)\}_{i=1}^d, q(Y|C_1, .., C_d)\right\}$:*

$$D_b(\hat{L}(\mathcal{Q})\|L(\mathcal{Q})) < \frac{\sum_{i=1}^d \left(n_i\tilde{I}(X_i; C_i) + m_i \ln n_i\right) + \left(\prod_{i=1}^d m_i\right) \ln |Y| + K}{N}, \tag{4.6}$$

where $K = \frac{1}{2}\ln(4N) - \ln \delta$.

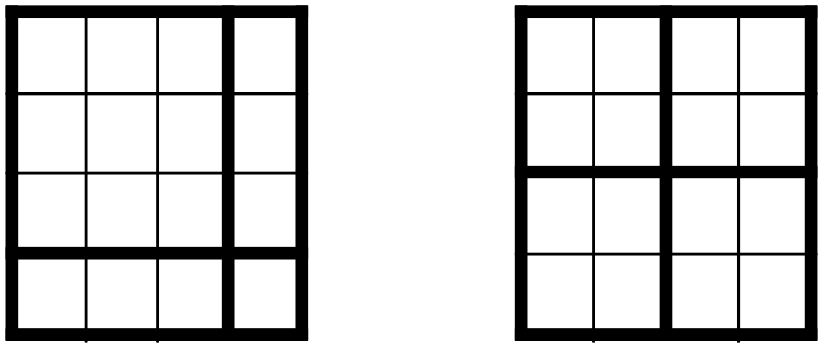
Remarks:

- Any bounded loss greater than 1 can be normalized to the [0,1] interval.
- All other forms of PAC-Bayesian bounds mentioned in section 3.3.5, as well as Occam's razor for randomized predictors (3.28) can be applied. In the case of Occam's razor, $\tilde{I}(X_i;C_i)$ in (4.6) is replaced by the entropy $H(\tilde{q}(c_i))$ of $\tilde{q}(c_i)$ as defined in (4.5) and the $\frac{1}{2}\ln(4N)$ factor is eliminated. In fact, in most situations Occam's razor suggests a tighter bound for this problem, but the PAC-Bayesian form is handier if optimization is required, because $\tilde{I}(X_i;C_i)$ has a convenient derivative with respect to $q(C_i|X_i)$.
- For purposes of the discussion below it is easier to look at the weaker, but explicit form of the bound (4.6), which follows from it by the L_1-norm lower bound on the KL-divergence (3.6):

$$L(\mathcal{Q}) < \hat{L}(\mathcal{Q}) + \sqrt{\frac{\sum_i \left(n_i \tilde{I}(X_i;C_i) + m_i \ln n_i\right) + \left(\prod_i m_i\right) \ln|Y| + K}{2N}}, \tag{4.7}$$

where $K = \frac{1}{2}\ln(4N) - \ln\delta$.

Discussion: There are two extreme solutions to collaborative filtering task that provide a good intuition on the co-clustering approach to this problem. If we assign all the data to a single large cluster we can evaluate the empirical mean/median/most frequent rating of that cluster fairly well. In this situation the empirical loss $\hat{L}(\mathcal{Q})$ is expected to be large, because we approximate all the entries with the global average, but its distance to the true loss $L(\mathcal{Q})$ is expected to be small. If we take the other extreme and assign each row and each column to a separate cluster, $\hat{L}(\mathcal{Q})$ can be zero, because we can approximate every entry with its own value, but its distance to the true loss $L(\mathcal{Q})$ is expected to be large, because each cluster has too little data to make a statistically reliable estimation. Thus, the goal is to optimize the tradeoff between locality of the predictions and their statistical reliability.



(a) Unbalanced Partition (b) Balanced Partition

Figure 4.2: **Illustration of an unbalanced (a) and a balanced (b) partitions of a 4×4 matrix into 2×2 clusters.** Note that there are 4 possible ways to group 4 objects into 2 unbalanced clusters and $\binom{4}{2} = 6$ possible ways to group 4 objects into 2 balanced clusters. Thus, the subspace of unbalanced partitions is smaller than the space of balanced partitions and unbalanced partitions are simpler (it is easier to describe an unbalanced partition rather than a balanced one).

One nice observation is that this tradeoff is explicitly exhibited in the bound in (4.7). If we assign all X_i-es to a single cluster, then $\tilde{I}(X_i; C_i) = 0$ and we obtain that $\hat{L}(\mathcal{Q})$ is close to $L(\mathcal{Q})$. And if we assign each X_i to a separate cluster, then $\tilde{I}(X_i; C_i)$ is large, specifically in this case $\tilde{I}(X_i; C_i) = \ln n_i$, and $\hat{L}(\mathcal{Q})$ is far from $L(\mathcal{Q})$. But there are even finer observations we can draw from this bound. Bear in mind that $n_i \tilde{I}(X_i; C_i)$ is linear in n_i, whereas $m_i \ln n_i$ is logarithmic in n_i. Thus, at least when m_i is small compared to n_i (which is a reasonable assumption when we cluster the values of X_i) the leading term in (4.7) is $n_i \tilde{I}(X_i; C_i)$. This term penalizes the *effective* complexity of the partition, rather than simply the number of clusters used in a solution. For example, the unbalanced partition of a 4×4 matrix into 2×2 clusters in Figure 4.2.a is simpler than the balanced partition into the same number of clusters in Figure 4.2.b. The reason, which will become clearer after we define the prior over the space of partitions in section 4.3, is that there are fewer unbalanced partitions than balanced ones. Slightly more

intuitively, the partition in Figure 4.2.a is closer to a partition where we put everything into one large cluster and does not fully utilize the 2×2 clusters it could use, and therefore should be penalized less. On a practical level, the bound enables at the optimization step to operate with more clusters than are actually required and to penalize the final solution according to the measure of utilization of the clusters. Thus, the bound (4.7) suggests a tradeoff between the empirical performance and the effective complexity of a partition.

Finally consider the $(\prod_{i=1}^{d} m_i) \ln |Y|$ term in the bound. $(\prod_{i=1}^{d} m_i)$ is the number of partition cells (in a hard partition) and it corresponds to the size of the $\langle C_1, .., C_d, Y \rangle$ clique in Figure 4.1.a. The number N of sample points should be comparable to the number of partition cells, so it is natural that this term appears in the bound. This term grows exponentially with the number of dimensions d, thus we can apply the bound for low-dimensional problems like collaborative filtering, but when the number of dimensions grows a different approach is required. We suggest one possible approach to handling high dimensions in Chapter 5.

Proof of Theorem 4.1. The proof of theorem 4.1 is a direct application of the PAC-Bayesian bound for classification in theorem 3.5. In order to apply the theorem we need to define a hypothesis space $\mathcal{H}$, a prior over hypothesis space $\mathcal{P}$, a posterior over hypothesis space $\mathcal{Q}$, and to calculate the KL-divergence $D(\mathcal{Q}\|\mathcal{P})$. We define the hypothesis space in the next section and design a prior over it in section 4.3. Then, substitution of the calculation of $D(\mathcal{Q}\|\mathcal{P})$ in lemma 4.2 into theorem 3.5 completes the proof. □

4.2 Grid Clustering Hypothesis Space

The hypothesis space $\mathcal{H}$ we chose to work with is the space of hard grid partitions of the product space $\mathcal{X}_1 \times .. \times \mathcal{X}_d$ (as illustrated in Figure 4.2) augmented with label assignments to the partition cells. (In section 4.4 we will use grid partitions without labels on the partition cells, thus the discussion in these two sections will deliberately be general enough to hold

in both cases.) In a hard grid partition each value $x_i \in \mathcal{X}_i$ is mapped to a single cluster $c_i \in \{1, .., m_i\}$. To work with $\mathcal{H}$ we use the following notations:

- We let $\bar{m} = (m_1, .., m_d)$ to be the vector counting the number of clusters along each dimension.
- We use $\mathcal{H}|_i$ to denote the space of partitions of $\mathcal{X}_i$. In other words, $\mathcal{H}|_i$ is a projection of $\mathcal{H}$ onto dimension i.
- We let $\mathcal{H}_{\bar{m}}$ denote the subspace of partitions of $\mathcal{X}_1 \times .. \times \mathcal{X}_d$ in which the number of clusters used along each dimension matches $\bar{m}$. Obviously, $\mathcal{H}_{\bar{m}}$-s are disjoint.
- We use $\mathcal{H}|_{y|\bar{m}}$ to denote the space of possible assignments of labels to $\mathcal{H}_{\bar{m}}$. Note that since the number of partition cells is equal for each member of $\mathcal{H}_{\bar{m}}$ there is one-to-one correspondence between $\mathcal{H}_{\bar{m}}$-s and $\mathcal{H}|_{y|\bar{m}}$-s. Thus, we can write $\mathcal{H} = \bigcup_{\bar{m}} \left(\mathcal{H}_{\bar{m}} \times \mathcal{H}|_{y|\bar{m}}\right)$.
- For each $h \in \mathcal{H}$ we write $h = h|_1 \times .. \times h|_d \times h|_{y|\bar{m}}$, where $h|_i$ denotes the partition induced by h along dimension i and $h|_{y|\bar{m}}$ denotes the assignment of labels to partition cells of h. Later, when we discuss density estimation with grid clustering, h is just $h = h|_1 \times .. \times h|_d$, without the labels assignment.

It should be pointed out that $\mathcal{Q} = \left\{\{q(C_i|X_i)\}_{i=1}^d, q(Y|C_1, .., C_d)\right\}$ is a distribution over $\mathcal{H}$ and (4.1) corresponds to a randomized prediction strategy. More precisely, $\mathcal{Q}$ is a distribution over $\mathcal{H}_{\bar{m}} \times \mathcal{H}|_{y|\bar{m}}$, where $\bar{m}$ matches the cardinalities of C_i-s in the definitions of $\left\{\{q(C_i|X_i)\}_{i=1}^d, q(Y|C_1, .., C_d)\right\}$. In order to draw a hypothesis $h \in \mathcal{H}$ according to $\mathcal{Q}$ we draw a cluster c_i for each $x_i \in \mathcal{X}_i$ according to $q(C_i|X_i)$ and then draw a label for each partition cell according to $q(Y|C_1, .., C_d)$. For example, we map each viewer to a cluster of movies, map each movie to a cluster of movies and assign ratings to the product space of viewer clusters by movie clusters. Then, in order to assign a label to a sample $\langle x_1, .., x_d \rangle$ we just check which partition cell it felt in and return the corresponding label. Recall that in order to assign a label to another sample point we have to draw a new hypothesis from $\mathcal{H}$.

Note that in (4.1) we actually skip the step of assigning a cluster for each $x_i \in \mathcal{X}_i$ and assigning a label for each partition cell (actually, the whole step of drawing a hypothesis) and assign a label to the given point $\langle X_1, .., X_d \rangle$ directly. Nevertheless, (4.1) corresponds to the randomized prediction process described above. This makes it possible to apply the PAC-Bayesian analysis.

4.3 Combinatorial Priors in PAC-Bayesian Bounds

In this section we design a combinatorial prior over the grid clustering hypothesis space and calculate the KL-divergence $D(\mathcal{Q}\|\mathcal{P})$ between the posterior defined earlier and the prior. An interesting point about the obtained result is that combinatorial priors result in mutual information terms in calculations of the KL-divergence. This can be compared with the L_2-norm and L_1-norm terms resulting from Gaussian and Laplacian priors respectively in the analysis of SVMs [69]. Another interesting point to mention is that the posterior $\mathcal{Q}$ returns a named partition of $\mathcal{X}_i$-s. However, the hypothesis space $\mathcal{H}$ and the prior $\mathcal{P}$ defined below operate with unnamed partitions: they only depend on the structure of a partition, but not on the exact names assigned to the clusters. This way we account for all possible name permutations, which are irrelevant for the solution.

The statements in the next two lemmas are given in two versions, one for the extended $\mathcal{H}$ with labels, which is used in the proofs of theorem 4.1, and the other one for the restricted version of $\mathcal{H}$ without the labels, which is used later for the proofs on density estimation with grid clustering.

Lemma 4.1. *It is possible to define a prior $\mathcal{P}$ over $\mathcal{H}_{\bar{m}}$ that satisfies*:

$$\mathcal{P}(h) \geq \frac{1}{\exp\left[\sum_{i=1}^{d} \left(n_i H(q_{h|_i}) + (m_i - 1)\ln n_i\right)\right]}, \tag{4.8}$$

where $q_{h|_i}$ denotes the cardinality profile of cluster sizes along dimension i of a partition corresponding to h. It is further possible to define a prior $\mathcal{P}$

over $\mathcal{H} = \bigcup_{\bar{m}} \left(\mathcal{H}_{\bar{m}} \times \mathcal{H}|_{y|\bar{m}}\right)$ *that satisfies*:

$$\mathcal{P}(h) \geq \frac{1}{\exp\left[\sum_{i=1}^{d}\left(n_i H(q_{h|_i}) + m_i \ln n_i\right) + \left(\prod_{i=1}^{d} m_i\right)\ln|Y|\right]}, \qquad (4.9)$$

Lemma 4.2. *For the prior defined in* (4.8) *and the posterior* $\mathcal{Q} = \{q(C_i|X_i)\}_{i=1}^{d}$:

$$D(\mathcal{Q}\|\mathcal{P}) \leq \sum_{i=1}^{d}\left(n_i \tilde{I}(X_i;C_i) + (m_i - 1)\ln n_i\right). \qquad (4.10)$$

And for the prior defined in (4.9) *and the posterior* $\mathcal{Q} = \left\{\{q(C_i|X_i)\}_{i=1}^{d}, q(Y|C_1,..,C_d)\right\}$:

$$D(\mathcal{Q}\|\mathcal{P}) \leq \sum_{i=1}^{d}\left(n_i \tilde{I}(X_i;C_i) + m_i \ln n_i\right) + \left(\prod_{i=1}^{d} m_i\right)\ln|Y|. \qquad (4.11)$$

4.3.1 Proofs

Proof of Lemma 4.1. To define the prior $\mathcal{P}$ over $\mathcal{H}_{\bar{m}}$ we count the hypotheses in $\mathcal{H}_{\bar{m}}$. There are $\binom{n_i-1}{m_i-1} \leq n_i^{m_i-1}$ possibilities to choose a cluster cardinality profile along a dimension i. (Each of the m_i clusters has a size of at least one. To define a cardinality profile we are free to distribute the "excess mass" of $n_i - m_i$ among the m_i clusters. The number of possible distributions equals the number of possibilities to place $m_i - 1$ ones in a sequence of $(n_i - m_i) + (m_i - 1) = n_i - 1$ ones and zeros.) For a fixed cardinality profile $q_{h|_i} = \{|c_{i1}|,..,|c_{im_i}|\}$ (over a single dimension) there are $\binom{n_i}{|c_{i1}|,..,|c_{im_i}|} \leq e^{n_i H(q_{h|_i})}$ possibilities to assign X_i-s to the clusters. Putting all the combinatorial calculations together we can define a distribution $\mathcal{P}(h)$ over $\mathcal{H}_{\bar{m}}$ that satisfies (4.8).

To prove (4.9) we further define a uniform prior over $\mathcal{H}|_{y|\bar{m}}$. Note that there are $|Y|^{\prod_i m_i}$ possibilities to assign labels to the partition cells in $\mathcal{H}_{\bar{m}}$. Finally, we define a uniform prior over the choice of $\bar{m}$. There are n_i possibilities to chose the value of m_i (we can assign all x_i-s to a single cluster, assign each x_i to a separate cluster, and all the possibilities in the middle). Combining this with the combinatorial calculations done for (4.8) suggests

(4.9) □

Proof of Lemma 4.2. We first handle the case with no labels. We use the decomposition $D(\mathcal{Q}\|\mathcal{P}) = -\mathbb{E}_{\mathcal{Q}}\mathcal{P}(h) - H(\mathcal{Q})$ and bound $-\mathbb{E}_{\mathcal{Q}}\mathcal{P}(h)$ and $H(\mathcal{Q})$ separately. We further decompose $\mathcal{P}(h) = \mathcal{P}(h|_1) \cdot .. \cdot \mathcal{P}(h|_d)$ and $\mathcal{Q}(h)$ in a similar manner. Then $-\mathbb{E}_{\mathcal{Q}} \ln \mathcal{P}(h) = -\sum_i \mathbb{E}_Q \ln \mathcal{P}(h|_i)$, and similarly for $D(\mathcal{Q}\|\mathcal{P})$. Therefore, we can treat each dimension separately.

To bound $-\mathbb{E}_{\mathcal{Q}} \ln \mathcal{P}(h|_i)$ recall that $\tilde{q}(c_i) = \frac{1}{n_i}\sum_{x_i} q(c_i|x_i)$ is the expected distribution over cardinalities of clusters along dimension i if we draw a cluster c_i for each value $x_i \in \mathcal{X}_i$ according to $q(C_i|X_i)$. Let $q_{h|_i}$ be a cluster cardinality profile obtained by such an assignment and corresponding to a hypothesis $h|_i$. Then by lemma 4.1:

$$-\mathbb{E}_{\mathcal{Q}} \ln \mathcal{P}(h|_i) \leq (m_i - 1)\ln n_i + n_i \mathbb{E}_{\tilde{q}(c_i)} H(q_{h|_i}). \tag{4.12}$$

To bound $\mathbb{E}_{\tilde{q}(c_i)} H(q_{h|_i})$ we use the result on the negative bias of empirical entropy estimates cited below. See [71] for a proof.

Theorem 4.2 (Paninski, 2003)**.** *Let $X_1, .., X_N$ be i.i.d. distributed by $p(X)$ and let $\hat{p}(X)$ be their empirical distribution. Then:*

$$\mathbb{E}_p H(\hat{p}) = H(p) - \mathbb{E}_p D(\hat{p}\|p) \leq H(p). \tag{4.13}$$

By (4.13) $\mathbb{E}_{\tilde{q}(c_i)} H(q_{h|_i}) \leq H(\tilde{q}(c_i))$. Substituting this into (4.12) yields:

$$-\mathbb{E}_{\mathcal{Q}} \ln \mathcal{P}(h|_i) \leq n_i H(\tilde{q}(c_i)) + (m_i - 1)\ln n_i. \tag{4.14}$$

Now we turn to compute $-H(\mathcal{Q}) = \mathbb{E}_{\mathcal{Q}} \ln \mathcal{Q}(h|_i)$. To do so we bound $\ln \mathcal{Q}(q_{h|_i})$ from above. The bound follows from the fact that if we draw n_i values of C_i according to $q(C_i|X_i)$ the probability of the resulting type is bounded from above by $e^{-n_i \tilde{H}(C_i|X_i)}$, where $\tilde{H}(C_i|X_i) = -\frac{1}{n_i}\sum_{x_i,c_i} q(c_i|x_i)\ln q(c_i|x_i)$ (see theorem 12.1.2 in [27]). Thus, $\mathbb{E}_{\mathcal{Q}} \ln \mathcal{Q}(h|_i) \leq -n_i \tilde{H}(C_i|X_i)$, which together with (4.14) and the identity $\tilde{I}(X_i; C_i) = H(\tilde{q}(c_i)) - \tilde{H}(C_i|X_i)$ completes the proof of (4.10).

To prove (4.11) we recall that $\mathcal{Q}$ is defined for a fixed $\bar{m}$. Hence,

$-\mathbb{E}_{\mathcal{Q}} \ln \mathcal{P}(h|_{y|\bar{m}}) = (\prod_{i=1}^{d} m_i) \ln |Y|$ and $-H(\mathcal{Q}(h|_{y|\bar{m}})) \leq 0$. Finally, by the choice of prior $\mathcal{P}(\bar{m})$ over the selection of $\bar{m}$ we have $-\mathbb{E}_{\mathcal{Q}} \ln \mathcal{P}(\bar{m}) = \sum_{i=1}^{d} \ln n_i$ and $H(\mathcal{Q}(\bar{m})) = 0$, which is added to (4.10) by the additivity of $D(\mathcal{Q}\|\mathcal{P})$ completing the proof. □

4.4 PAC-Bayesian Analysis of Density Estimation with Grid Clustering

In this section we derive a generalization bound for density estimation with grid clustering. This time we have no labels and the goal is to find a good estimator for an unknown joint probability distribution $p(X_1, .., X_d)$ over a d-dimensional product space $\mathcal{X}_1 \times .. \times \mathcal{X}_d$ based on a sample of size N from p. As an illustrative example, think of estimating a joint probability distribution of words and documents (X_1 and X_2) from their co-occurrence matrix. The goodness of an estimator q for p is measured by $-\mathbb{E}_{p(X_1,..,X_d)} \ln q(X_1, .., X_d)$.

By theorem 3.8, to obtain a meaningful bound for a direct estimation of $p(X_1, .., X_d)$ we need N to be exponential in n_i-s, since the cardinality of the random variable $\langle X_1, .., X_d \rangle$ is $\prod_i n_i$. To reduce this dependency to be linear in $\sum_i n_i$ we restrict the estimator $q(X_1, .., X_d)$ to be of the factor form:

$$\begin{aligned} q(X_1, .., X_d) &= \sum_{C_1,..,C_d} q(C_1, .., C_d) \prod_{i=1}^{d} q(X_i|C_i) \\ &= \sum_{C_1,..,C_d} q(C_1, .., C_d) \prod_{i=1}^{d} \frac{q(X_i)}{q(C_i)} q(C_i|X_i). \end{aligned} \tag{4.15}$$

We emphasize that the above decomposition assumption is only on the estimator q, but not on the generating distribution p.

We choose the hypothesis space $\mathcal{H}$ to be the space of hard partitions of the product space $\mathcal{X}_1 \times .. \times \mathcal{X}_d$, as previously; however this time there are no labels to the partition cells. The general message of the following theorems is that the empirical distribution over the coarse partitioned space converges to the true one, and then we can use (4.15) to extrapolate it back on the

whole space $\mathcal{X}_1 \times .. \times \mathcal{X}_d$. Next we state this more formally.

We recall from the previous section that a distribution $\mathcal{Q} = \{q(C_i|X_i)\}_{i=1}^d$ is a distribution over $\mathcal{H}_{\bar{m}}$. To obtain a hypothesis $h \in \mathcal{H}_{\bar{m}}$ we draw a cluster for each $x_i \in \mathcal{X}_i$ according to $q(C_i|X_i)$. The way we have written (4.15) enables us to view it as a randomized prediction process: we draw a hypothesis h according to $\mathcal{Q}$ and then predict the probability of $\langle X_1, .., X_d \rangle$ as $q(C_1^h(X_1), .., C_d^h(X_d)) \prod_i \frac{q(X_i)}{q(C_i^h(X_i))}$, where $C_i^h(X_i) = h(X_i)$ is the partition cell that X_i fell within in h. Although (4.15) skips the process of drawing the complete partition h and returns the probability of $\langle X_1, .., X_d \rangle$ directly, the described randomized prediction process matches the predictions done by (4.15) and thus enables us to apply the PAC-Bayesian bounds.

Let $h \in \mathcal{H}$ be a hard partition of $\mathcal{X}_1 \times .. \times \mathcal{X}_d$ and let $h(x_i)$ denote the cluster that x_i is mapped to in h. We define the distribution over the partition cells $\langle C_1, .., C_d \rangle$ induced by p and h:

$$p_h(c_1, .., c_d) = \sum_{\substack{x_1,..,x_d: \\ h(x_i)=c_i}} p(x_1, .., x_d), \tag{4.16}$$

$$p_h(c_i) = \sum_{x_i : h(x_i)=c_i} p(x_i). \tag{4.17}$$

We further define the distribution over the partition cells induced by h and the empirical distribution $\hat{p}(X_1, .., X_d)$ corresponding to the sample by substitution of $\hat{p}$ instead of p in the above definitions:

$$\hat{p}_h(c_1, .., c_d) = \sum_{\substack{x_1,..,x_d: \\ h(x_i)=c_i}} \hat{p}(x_1, .., x_d), \tag{4.18}$$

$$\hat{p}_h(c_i) = \sum_{x_i : h(x_i)=c_i} \hat{p}(x_i). \tag{4.19}$$

We also define the distribution over partition cells induced by $\mathcal{Q}$ and p:

$$p_{\mathcal{Q}}(c_1, .., c_d) = \sum_h \mathcal{Q}(h) p_h(c_1, .., c_d)$$

$$= \sum_{x_1,..,x_d} p(x_1,..,x_d) \prod_{i=1}^{d} q(c_i|x_i), \tag{4.20}$$

$$p_{\mathcal{Q}}(c_i) = \sum_h \mathcal{Q}(h) p_h(c_i) = \sum_{x_i} p(x_i) q(c_i|x_i). \tag{4.21}$$

And its empirical counterpart:

$$\hat{p}_{\mathcal{Q}}(c_1,..,c_d) = \sum_h \mathcal{Q}(h) \hat{p}_h(c_1,..,c_d)$$

$$= \sum_{x_1,..,x_d} \hat{p}(x_1,..,x_d) \prod_{i=1}^{d} q(c_i|x_i), \tag{4.22}$$

$$\hat{p}_{\mathcal{Q}}(c_i) = \sum_h \mathcal{Q}(h) \hat{p}_h(c_i) = \sum_{x_i} \hat{p}(x_i) q(c_i|x_i). \tag{4.23}$$

We extend the definitions of p_h, $p_{\mathcal{Q}}$, $\hat{p}_h$ and $\hat{p}_{\mathcal{Q}}$ for the whole space $\mathcal{X}_1 \times .. \times \mathcal{X}_d$ using (4.15). We use the notation $C_i^h(X_i) = h(X_i)$ to denote the cluster $C_i^h(X_i)$ that X_i is mapped to by h.

$$p_h(X_1,..,X_d) = p_h(h(X_1),..,h(X_d)) \prod_{i=1}^{d} \frac{p(X_i)}{p_h(h(X_i))}$$

$$= p_h(C_1^h(X_1),..,C_d^h(X_d)) \prod_{i=1}^{d} \frac{p(X_i)}{p_h(C_i^h(X_i))}, \tag{4.24}$$

$$p_{\mathcal{Q}}(X_1,.,X_d) = \sum_{C_1,.,C_d} p_{\mathcal{Q}}(C_1,.,C_d) \prod_{i=1}^{d} \frac{p(X_i)}{p_{\mathcal{Q}}(C_i)} q(C_i|X_i), \tag{4.25}$$

$$\hat{p}_h(X_1,..,X_d) = \hat{p}_h(h(X_1),..,h(X_d)) \prod_{i=1}^{d} \frac{\hat{p}(X_i)}{\hat{p}_h(h(X_i))}$$

$$= \hat{p}_h(C_1^h(X_1),..,C_d^h(X_d)) \prod_{i=1}^{d} \frac{\hat{p}(X_i)}{\hat{p}_h(C_i^h(X_i))}, \tag{4.26}$$

$$\hat{p}_{\mathcal{Q}}(X_1,.,X_d) = \sum_{C_1,.,C_d} \hat{p}_{\mathcal{Q}}(C_1,.,C_d) \prod_{i=1}^{d} \frac{\hat{p}(X_i)}{\hat{p}_{\mathcal{Q}}(C_i)} q(C_i|X_i). \tag{4.27}$$

Note that $p_{\mathcal{Q}}(X_1,..,X_d)$ is the distribution over $\mathcal{X}_1 \times .. \times \mathcal{X}_d$, which has the

form (4.15) and is the closest to the true distribution $p(X_1, .., X_d)$ under the constraint that $\{q(C_i|X_i)\}_{i=1}^d$ are fixed. Further, note that since we have no access to $p(X_1, .., X_d)$ we do not know $p_{\mathcal{Q}}(X_1, .., X_d)$. In the next theorem we state that the distributions $\hat{p}_{\mathcal{Q}}(X_1, .., X_d)$, $\hat{p}_{\mathcal{Q}}(C_1, .., C_d)$, and $\hat{p}_{\mathcal{Q}}(X_i)$ based on the sample converge to their counterparts corresponding to the true distribution $p(X_1, .., X_d)$.

Theorem 4.3. *For any probability measure p over $\mathcal{X}_1 \times .. \times \mathcal{X}_d$ and an i.i.d. sample S of size N according to p, with a probability of at least $1 - \delta$ for all grid clusterings $\mathcal{Q} = \{q(C_i|X_i)\}_{i=1}^d$ the following holds simultaneously*:

$$D(\hat{p}_{\mathcal{Q}}(C_1, ., C_d) \| p_{\mathcal{Q}}(C_1, ., C_d)) \leq \frac{\sum_{i=1}^d n_i \tilde{I}(X_i; C_i) + K_1}{N} \tag{4.28}$$

and

$$D(\hat{p}(X_i) \| p(X_i)) \leq \frac{(n_i - 1)\ln(N+1) + \ln\frac{d+1}{\delta}}{N}, \tag{4.29}$$

where

$$K_1 = \sum_{i=1}^d m_i \ln n_i + (M-1)\ln(N+1) + \ln\frac{d+1}{\delta} \tag{4.30}$$

and

$$M = \prod_{i=1}^d m_i \tag{4.31}$$

is the number of partition cells in $\mathcal{Q}$.

As well, with a probability greater than $1 - \delta$:

$$D(\hat{p}_{\mathcal{Q}}(X_1, ., X_d) \| p_{\mathcal{Q}}(X_1, ., X_d)) \leq \frac{\sum_{i=1}^d n_i \tilde{I}(X_i; C_i) + K_2}{N}, \tag{4.32}$$

where

$$K_2 = \sum_i m_i \ln n_i + \left[M + \sum_i n_i - d - 1\right] \ln(N+1) - \ln\delta. \tag{4.33}$$

Before we prove and discuss the theorem we point out that although $\hat{p}_{\mathcal{Q}}(X_1, ., X_d)$ converges to $p_{\mathcal{Q}}(X_1, ., X_d)$ it still cannot be used to minimize

$-\mathbb{E}_{p(X_1,..,X_d)} \ln \hat{p}_{\mathcal{Q}}(X_1,..,X_d)$, because it is not bounded from zero. Similarly we cannot use the smoothing theorem 3.9 to smooth $\hat{p}_{\mathcal{Q}}(X_1,..,X_d)$ directly, because the cardinality of the random variable $\langle X_1,..,X_d\rangle$ is $\prod_i n_i$ and this term will enter into the bounds. To get around this we utilize the factor form of p_h and the bounds (4.28) and (4.29). We define an estimator $q_{\mathcal{Q}}$, which is a smoothed version of $\hat{p}_{\mathcal{Q}}$ in the following way:

$$q_h(C_1,..,C_d) = \frac{\hat{p}_h(C_1,..,C_d)+\gamma}{1+\gamma M}, \tag{4.34}$$

$$q(X_i) = \frac{\hat{p}(X_i)+\gamma_i}{1+\gamma_i n_i}, \tag{4.35}$$

$$q_h(c_i) = \sum_{x_i:h(x_i)=c_i} q(x_i), \tag{4.36}$$

$$q_h(X_1,..,X_d) = q_h(C_1^h(X_1),..,C_d^h(X_d)) \prod_{i=1}^{d} \frac{q_h(X_i)}{q_h(C_i^h(X_i))}. \tag{4.37}$$

And for a distribution $\mathcal{Q}$ over $\mathcal{H}$:

$$q_{\mathcal{Q}}(C_1,..,C_d) = \frac{\hat{p}_{\mathcal{Q}}(C_1,..,C_d)+\gamma}{1+\gamma M}, \tag{4.38}$$

$$q_{\mathcal{Q}}(C_i) = \sum_{x_i} q(x_i)q(C_i|x_i) = \frac{\hat{p}_{\mathcal{Q}}(C_i)+\gamma_i \tilde{q}(C_i) n_i}{1+\gamma_i n_i}, \tag{4.39}$$

$$\begin{aligned} q_{\mathcal{Q}}(X_1,..,X_d) &= \sum_h \mathcal{Q}(h) q_h(X_1,..,X_d) \\ &= \sum_{C_1,..,C_d} q_{\mathcal{Q}}(C_1,..,C_d) \prod_{i=1}^{d} \frac{q(X_i)}{q_{\mathcal{Q}}(C_i)} q(C_i|X_i). \end{aligned} \tag{4.40}$$

In the following theorem we provide a bound on $-\mathbb{E}_{p(X_1,..,X_d)} \ln q_{\mathcal{Q}}(X_1,..,X_d)$. Note, that we take the expectation with respect to the true, unknown distribution p that may have an arbitrary form.

Theorem 4.4. *For the density estimator $q_{\mathcal{Q}}(X_1,..,X_d)$ defined by equations* (4.35)*,* (4.38)*,* (4.39)*, and* (4.40)*, $-\mathbb{E}_{p(X_1,..,X_d)} q_{\mathcal{Q}}(X_1,..,X_d)$ attains its minimum at $\gamma = \frac{\sqrt{\varepsilon/2}}{M}$ and $\gamma_i = \frac{\sqrt{\varepsilon_i/2}}{n_i}$, where ε is defined by the right-hand side of* (4.28) *and ε_i is defined by the right-hand side of* (4.29)*. At*

this optimal level of smoothing, with a probability greater than $1-\delta$ *for all* $\mathcal{Q}=\{q(C_i|X_i)\}_{i=1}^d$:

$$-\mathbb{E}_{p(X_1,..,X_d)}\ln q_{\mathcal{Q}}(X_1,..,X_d) \leq -I(\hat{p}_{\mathcal{Q}}(C_1,..,C_d))+\ln(M)\sqrt{\frac{\sum_{i=1}^d n_i\tilde{I}(X_i;C_i)+K_1}{2N}}+\phi(\varepsilon)+K_3, \tag{4.41}$$

where $I(\hat{p}_{\mathcal{Q}}(C_1,..,C_d))=\left[\sum_{i=1}^d H(\hat{p}_{\mathcal{Q}}(C_i))\right]-H(\hat{p}_{\mathcal{Q}}(C_1,..,C_d))$ *is the multi-information between* $C_1,..,C_d$ *with respect to* $\hat{p}_{\mathcal{Q}}(C_1,..,C_d)$*,*

$$K_3=\left[\sum_{i=1}^d H(\hat{p}(X_i))+2\sqrt{\varepsilon_i/2}\ln n_i+\phi(\varepsilon_i)+\psi(\varepsilon_i)\right],$$

and the functiond ϕ *and* ψ *are defined in theorem 3.9.*

Discussion: We discuss theorem 4.4 first. We point out that $q_{\mathcal{Q}}(X_1,..,X_d)$ is directly related to $\hat{p}_{\mathcal{Q}}(X_1,..,X_d)$ and that $\hat{p}_{\mathcal{Q}}(X_1,..,X_d)$ is determined by the empirical frequencies $\hat{p}(X_1,..,X_d)$ of the sample and our choice of $\mathcal{Q}=\{q(C_i|X_i)\}_{i=1}^d$. There are only two quantities in the bound (4.41) that depend on our choice of $\mathcal{Q}$; they are: $-I(\hat{p}_{\mathcal{Q}}(C_1,..,C_d))$ and $\sum_i \frac{n_i}{N}\tilde{I}(X_i;C_i)$ [note that the latter also appears in $\phi(\varepsilon)$]. Thus, theorem 4.4 suggests that a good estimator $q_{\mathcal{Q}}(X_1,..X_d)$ of $p(X_1,..,X_d)$ should optimize the trade-off between $-I(\hat{p}_{\mathcal{Q}}(C_1,..,C_d))$ and $\sum_i \frac{n_i}{N}\tilde{I}(X_i;C_i)$. Similar to theorem 4.1, the latter corresponds to the mutual information that the hidden cluster variables preserve on the observed variables. Larger values of $\tilde{I}(X_i;C_i)$ correspond to partitions of $\mathcal{X}_1,..,\mathcal{X}_d$, which are more complex. The first term, $-I(\hat{p}_{\mathcal{Q}}(C_1,..,C_d))$, corresponds to the amount of structural information on C_i-s extracted by the partition. More precisely, we should look at the value of $\sum_i H(\hat{p}(X_i))-I(\hat{p}_{\mathcal{Q}}(C_1,..,C_d))$, where $\sum_i H(\hat{p}(X_i))$ is a part of K_3 and roughly corresponds to the performance we can obtain by approximating $p(X_1,..,X_d)$ with a product of empirical marginals $\hat{p}(X_1)\cdot..\cdot\hat{p}(X_d)$. Thus $-I(\hat{p}_{\mathcal{Q}}(C_1,..,C_d))$ is the added value of the partition in estimating $p(X_1,..,X_d)$. Since $\sum_i H(\hat{p}(X_i))\geq I(\hat{p}_{\mathcal{Q}}(C_1,..,C_d))$ the bound 4.41 is al-

ways positive.

The value of $I(\hat{p}_{\mathcal{Q}}(C_1, .., C_d))$ increases monotonically with the increase of the partition complexity $\mathcal{Q}$ (we can see this by the information processing inequality [27]). Thus, the tradeoff in (4.41) is analogical to the tradeoff in (4.6): the partition $\mathcal{Q}$ should balance its utility function $-I(\hat{p}_{\mathcal{Q}}(C_1, .., C_d))$ and the statistical reliability of the estimate of the utility function, which is related to $\sum_i \frac{n_i}{N} \tilde{I}(X_i; C_i)$. This tradeoff suggests a modification to the original objective of co-clustering in [32], which is maximization of $I(C_1; C_2)$ alone (Dhillon et. al. [32] discuss the case of two-dimensional matrices). The tradeoff in (4.41) can be applied to model order selection.

Now we make a few comments about theorem 4.3. An interesting point about this theorem is that the cardinality of the random variable $\langle X_1, .., X_d \rangle$ is $\prod_i n_i$. Thus, a direct application of theorem 3.6 to bound $D(\hat{p}_{\mathcal{Q}}(X_1, ., X_d) \| p_{\mathcal{Q}}(X_1, ., X_d))$ will insert this term into the bound. However, by using the factor form (4.15) of $\hat{p}_{\mathcal{Q}}(X_1, ., X_d)$ and $p_{\mathcal{Q}}(X_1, ., X_d)$ we are able to reduce this dependency to $M + \sum_i n_i - d - 1$. This result hints at the great potential of applying PAC-Bayesian analysis to more complex graphical models. Exploration of this direction should be a topic for future work.

4.4.1 Proofs

We conclude this section by presenting the proofs of theorems 4.3 and 4.4.

Proof of Theorem 4.3. The proof is based on the PAC-Bayesian theorem 3.6 on density estimation. To apply the theorem we need to define a prior $\mathcal{P}$ over $\mathcal{H}$ and then calculate $D(\mathcal{Q} \| \mathcal{P})$. We note that for a fixed $\mathcal{Q}$ the cardinalities of the clusters $\bar{m}$ are fixed. There are $\prod_i n_i$ disjoint subspaces $\mathcal{H}_{\bar{m}}$ in $\mathcal{H}$. We handle each $\mathcal{H}_{\bar{m}}$ independently and then combine the results to obtain theorem 4.3.

By theorem 3.6 and lemma 4.2, for the prior $\mathcal{P}$ over $\mathcal{H}_{\bar{m}}$ defined in lemma 4.1, with a probability greater than $1 - \frac{\delta}{(d+1)\prod_i n_i}$ we obtain (4.28) for each $\mathcal{H}_{\bar{m}}$. In addition, by theorem 3.8 with a probability greater than $1 - \frac{\delta}{d+1}$ inequality (4.29) holds for each X_i. By a union bound over the $\prod_i n_i$

subspaces of $\mathcal{H}$ and the d variables X_i we obtain that (4.28) and (4.29) hold simultaneously for all $\mathcal{Q}$ and X_i with a probability greater than $1-\delta$.

To prove (4.32), fix some hard partition h and let $C_i^h = h(X_i)$. Then:

$$\begin{aligned}
&D(\hat{p}_h(X_1,..,X_d)\|p_h(X_1,..,X_d)) \\
&\quad = D(\hat{p}_h(X_1,..,X_d,C_1^h(X_1),..,C_d^h(X_d))\|p_h(X_1,..,X_d,C_1^h(X_1),..,C_d^h(X_d))) \\
&\quad = D(\hat{p}_h(C_1,..,C_d)\|p_h(C_1,..,C_d)) \\
&\qquad + D(\hat{p}_h(X_1,..,X_d|C_1^h(X_1),..,C_d^h(X_d))\|p_h(X_1,..,X_d|C_1^h(X_1),..,C_d^h(X_d))) \\
&\quad = D(\hat{p}_h(C_1,..,C_d)\|p_h(C_1,..,C_d)) + \sum_{i=1}^{d} D(\hat{p}_h(X_i|C_i^h(X_i))\|p_h(X_i|C_i^h(X_i))) \\
&\quad = D(\hat{p}_h(C_1,..,C_d)\|p_h(C_1,..,C_d)) + \sum_{i=1}^{d} D(\hat{p}(X_i)\|p(X_i)) \\
&\qquad - \sum_{i=1}^{d} D(\hat{p}_h(C_i)\|p_h(C_i)) \\
&\quad \leq D(\hat{p}_h(C_1,..,C_d)\|p_h(C_1,..,C_d)) + \sum_{i=1}^{d} D(\hat{p}(X_i)\|p(X_i)).
\end{aligned}$$

And:

$$\begin{aligned}
&\mathbb{E}_S e^{ND(\hat{p}_h(X_1,..,X_d)\|p_h(X_1,..,X_d))} \\
&\qquad \leq \left(\mathbb{E}_S e^{ND(\hat{p}_h(C_1,..,C_d)\|p_h(C_1,..,C_d))}\right) \prod_{i=1}^{d} \mathbb{E}_S e^{ND(\hat{p}(X_i)\|p(X_i))} \\
&\qquad \leq (N+1)^{M+\sum_{i=1}^{d} n_i-(d+1)},
\end{aligned}$$

where the last inequality is by theorem 3.7. From here we follow the lines of the proof of theorem 3.6. Namely:

$$\begin{aligned}
\mathbb{E}_S\left[\mathbb{E}_{\mathcal{P}(h)} e^{ND(\hat{p}_h(X_1,..,X_d)\|p_h(X_1,..,X_d))}\right] &= \mathbb{E}_{\mathcal{P}(h)}\left[\mathbb{E}_S e^{ND(\hat{p}_h(X_1,..,X_d)\|p_h(X_1,..,X_d))}\right] \\
&\leq (N+1)^{M+\sum_{i=1}^{d} n_i-(d+1)}.
\end{aligned}$$

Thus, by Markov's inequality $\mathbb{E}_{\mathcal{P}(h)} e^{ND(\hat{p}_h(X_1,..,X_d)\|p_h(X_1,..,X_d))} \leq \frac{1}{\delta}(N+$

$1)^{M+\sum_i n_i-(d+1)}$ with a probability of at least $1-\delta$ and (4.32) follows by the change of measure inequality (3.38) and convexity of the KL-divergence, when the prior $\mathcal{P}$ over $\mathcal{H}$ defined in lemma 4.1 is selected (this time we give a weight of $(\prod_i n_i)^{-1}$ to each $\mathcal{H}_{\bar{m}}$ and obtain a prior over the whole $\mathcal{H}$). The calculation of $D(\mathcal{Q}\|\mathcal{P})$ for this prior is done in lemma 4.2. □

Proof of Theorem 4.4.

$$\begin{aligned}
-\mathbb{E}_{p(X_1,..,X_d)} \ln q_{\mathcal{Q}}(X_1,..,X_d) &= -\mathbb{E}_{p(X_1,..,X_d)} \ln \mathbb{E}_{\mathcal{Q}(h)} q_h(X_1,..,X_d) \\
&\leq -\mathbb{E}_{\mathcal{Q}(h)} \mathbb{E}_{p(X_1,..,X_d)} \ln q_h(X_1,..,X_d) \\
&= -\mathbb{E}_{\mathcal{Q}(h)} \mathbb{E}_{p(X_1,..,X_d)} \ln q_h(C_1^h(X_1),..,C_d^h(X_d)) \prod_i \frac{q(X_i)}{q_h(C_i^h(X_i))} \\
&= -\mathbb{E}_{\mathcal{Q}(h)} [\mathbb{E}_{p_h(C_1,..,C_d)} \ln q_h(C_1,..,C_d)] - \sum_i \mathbb{E}_{p(X_i)} \ln q(X_i) \\
&\quad + \sum_i \mathbb{E}_{\mathcal{Q}(h)} \mathbb{E}_{p_h(C_i)} \ln q_h(C_i) \\
&\leq -\mathbb{E}_{\mathcal{Q}(h)} [\mathbb{E}_{p_h(C_1,..,C_d)} \ln q_h(C_1,..,C_d)] - \sum_i \mathbb{E}_{p(X_i)} \ln q(X_i) \\
&\quad + \sum_i \mathbb{E}_{p_{\mathcal{Q}}(C_i)} \ln q_{\mathcal{Q}}(C_i)
\end{aligned}$$

At this point we use (3.47) to bound the first and the second term and the lower bound (3.49) to bound the last term and obtain (4.41). □

Chapter 5

Beyond Co-clustering

The analysis of co-clustering presented in the previous chapter holds for any dimension d. However, the dependence of the bounds (4.6), (4.32), and (4.41) on d is exponential because of the $M = \prod_i m_i$ term they involve. Thus, high dimensional tasks require a different treatment. Some improvements are also possible if we consider discriminative prediction based on a single parameter X (i.e., in the case of $d = 1$). In this chapter we first consider the case of $d = 1$ and then the case of $d > 2$. The analysis of high dimensional cases suggests a possibility of PAC-Bayesian analysis of graphical models. This is further discussed in section 5.3.

5.1 Optimal Solution for One Dimension ($d = 1$)

In this section we prove that in classification by a single parameter there is no need for intermediate clustering of the parameter values. Instead of clustering the values of X we apply a much simpler procedure of smoothing the empirical conditional probabilities $\hat{p}(Y|X)$. It is proved here that such smoothing can produce at least as good a classifier $q(Y|X)$ as the best possible clustering. Furthermore, we show that it is possible to find the globally optimal form of smoothing (at least, from the point of view of its generalization properties in predicting Y according to the PAC-Bayesian bounds). In the applications section we demonstrate that the obtained bound is ex-

tremely tight and is less than 10% away from the test error.

One possible application we suggest for our bound is feature ranking. As opposed to the frequently applied practice of ranking features according to their mutual information or correlation with the label, the bound enables ranking of features according to their generalization potential. This is especially important when the cardinalities n_i-s of the features differ significantly. For example, if want to predict the probability of a person developing cancer based on his age (X_1) or based on the binary feature of whether the person smokes or not (X_2), the age variable is likely to have a higher correlation with the label just because it has more values, but this does not mean it has better predictive abilities. The experiments presented here support the advantage of the bound as a tool for feature rating.

We call the prediction rules $q(Y|X)$ that map the parameter (feature) values X directly to the label Y *direct mappings*. To prove the superiority of direct mappings over clustering-based solutions we start with the observation that for any clustering $\mathcal{Q}_c = \{q(C|X), q(Y|C)\}$ a classification rule $q(Y|X)$ defined as

$$q(y|x) = \sum_c q(y|c)q(c|x) \tag{5.1}$$

achieves the same loss as the loss of $\mathcal{Q}_c$. Therefore, the space of all direct mappings $q(Y|X)$ incorporates all possible solutions $\mathcal{Q}_c = \{q(C|X), q(Y|C)\}$ that can be achieved via intermediate clustering. It remains to be shown that the generalization power of the direct mappings is not worse than the generalization power of clustering-based solutions and that the global optimum can be found efficiently.

To analyze the generalization power of direct mappings we define $|Y|$ clusters c_y, one for each label $y \in \mathcal{Y}$, i.e., $C_y = \{c_y : y \in \mathcal{Y}\}$. We further observe that the prediction rule (5.1) corresponds to the prediction strategy $\mathcal{Q}_y = \{q(C_y|X), q(Y|C_y)\}$, where $q(y|c_{y'}) = \delta(y, y')$, where $\delta(y, y')$ is the Kronecker delta, and $q(c_y|x) = q(y|x)$, where $q(y|x)$ is defined by (5.1). In other words, the clustering C_y is identified with the labeling Y. We can apply the PAC-Bayesian bound (4.6) to bound the prediction ability of $\mathcal{Q}_y$. Since C_y is identified with Y we can replace $\tilde{I}(X;C)$ in (4.6) with $\tilde{I}(X;Y)$,

where

$$\tilde{I}(X;Y) = \frac{1}{n}\sum_{x,y} q(y|x) \ln \frac{q(y|x)}{\tilde{q}(y)}, \tag{5.2}$$

where $n = |X|$ and

$$\tilde{q}(y) = \frac{1}{n}\sum_{x} q(y|x). \tag{5.3}$$

Note that $q(y|x)$ is our prediction strategy and it is known; thus all the above quantities are known and depend on $q(y|x)$. By the information processing inequality [27], $\tilde{I}(X;C) \geq \tilde{I}(X;Y)$ for $\tilde{I}(X;C)$ corresponding to $\mathcal{Q}_c$. Furthermore, since the predictions of $\mathcal{Q}_c$ are identical to the predictions of the corresponding $\mathcal{Q}_y$ their empirical losses are equal. Thus the bound (4.6) for the direct mapping $\mathcal{Q}_y$ is at least as tight as the same bound for the clustering-based mapping $\mathcal{Q}_c$ to which it corresponds. Therefore, we can do the optimization of the prediction rule within the space of direct mappings only without compromising for the quality. We will show later on that it is possible to find globally optimal prediction rule within the space of direct mappings.

We can tighten the bound further by realizing that in the space of direct mappings each label is assigned to exactly one cluster. Thus there are exactly $|Y|$ clusters and at most $(|Y|!)$ possibilities to assign labels to these clusters (instead of n possibilities to choose the number of clusters and $m^{|Y|}$ possibilities to assign them labels in a general grid clustering). Thus, (4.6) can be improved in this case to:

$$D_b(\hat{L}(\mathcal{Q}_y)\|L(\mathcal{Q}_y)) < \frac{n\tilde{I}(X;Y) + \ln\left[\binom{n+|Y|-1}{|Y|-1}\right] + \ln(|Y|!) + \frac{1}{2}\ln(4N) - \ln\delta}{N} \tag{5.4}$$

$$\leq \frac{n\tilde{I}(X;Y) + (|Y|-1)\ln(n+1) + |Y|\ln|Y| - |Y| + \frac{1}{2}\ln(4N) - \ln\delta}{N}$$

Note that the bound (5.4) holds for any prediction rule $q(Y|X)$, no matter how it was obtained. Furthermore, it is possible to find the global optimum of (5.4). In order to find the global optimum observe that $L(\mathcal{Q}_y)$ depends on the tradeoff between $\hat{L}(\mathcal{Q}_y)$ and $\tilde{I}(X;Y)$ and that $\hat{L}(\mathcal{Q}_y)$ is

linear in $q(Y|X)$ and $\tilde{I}(X;Y)$ is convex in $q(Y|X)$. Thus, we can minimize the parameterized tradeoff $\hat{L}(\mathcal{Q}_y) + \beta\tilde{I}(X;Y)$ and apply a linear search over β to find the globally optimal bound on $L(\mathcal{Q}_y)$.

It is also possible to apply Occam's razor bound (3.28) for randomized classifiers to obtain a bound on $L(\mathcal{Q}_y)$. By Occam's razor bound we obtain:

$$D_b(\hat{L}(\mathcal{Q}_y)\|L(\mathcal{Q}_y)) < \frac{nH(\tilde{q}(Y)) + \ln\left[\binom{n+|Y|-1}{|Y|-1}\right] + \ln(|Y|!) - \ln\delta}{N}. \quad (5.5)$$

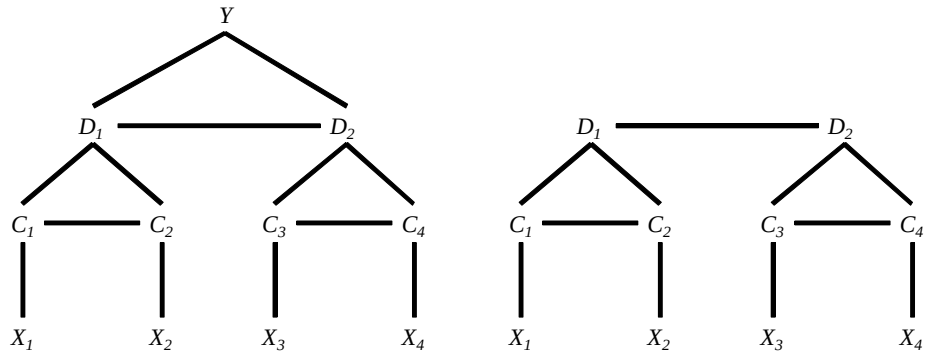
Unfortunately, (5.5) cannot be optimized with a gradient descent since $H(\tilde{q}(Y))$ has no derivative with respect to $q(Y|X)$, but if applied to the maximum likelihood prediction rule, in practice it often suggests a slightly tighter bound than the optimization of (5.4) [e.g., for the zero-one loss the maximum likelihood prediction rule is $q_{ml}(x) = \arg\max_y \hat{p}(y|x)$].

A related bound on generalization in prediction by a single feature was proposed in [74, 75]. Sabato and Shalev-Shwartz designed an estimator for the loss of a prediction rule based on the empirical frequencies $q_{emp}(y|x) = \hat{p}(y|x)$. They proved that their estimate is at most $O\left(\frac{\ln(N/\delta)\sqrt{\ln(1/\delta)}}{\sqrt{N}}\right)$ far from the generalization error of q_{emp}. Compared to their work, a strong advantage of the bound in (5.4) is that it holds for any prediction rule $q(Y|X)$. In particular, it holds for the maximum likelihood prediction $q_{ml}(x) = \arg\max_y \hat{p}(y|x)$ that performs much better than q_{emp} in practice.

Note that direct mapping is no longer optimal when there is more than one parameter. For example, for two parameters X_1, X_2 with cardinalities n_1, n_2 direct smoothing of the empirical conditional probability $\hat{p}(Y|X_1, X_2)$ introduces the $(n_1 n_2)\tilde{I}(\langle X_1, X_2\rangle, Y)$ term into the bound, whereas in a clustering-based solution this term is decomposed into $\sum_{i=1}^{2} n_i\tilde{I}(X_i; C_i)$.

5.2 High Dimensions ($d > 2$)

In this section we suggest a possible approach to analysis of high dimensional tasks, where the parameter space $\mathcal{X}_1, .., \mathcal{X}_d$ is more than 2-dimensional ($d > 2$). Recall that the bounds (4.6), (4.32), and (4.41) all involve the number of



(a) Discriminative Prediction (b) Density Estimation

Figure 5.1: **Illustration of graphical models for discriminative prediction and density estimation in high-dimensional spaces.** In both illustrations $d = 4$.

partition cells $M = \prod_i m_i$. This term is reasonably small when the number of dimensions is small (two or three). However, as the number of dimensions grows, this term grows exponentially. For example, if $d = 10$ and we use only 2 clusters along each of the 10 dimensions, this already yields $2^{10} = 1024$ partition cells.

One possible way to handle high dimensional cases is to use hierarchical partitions, as shown in Figure 5.1. For example, the discriminative prediction model corresponding to the model in Figure 5.1.a is:

$$q(Y|X_1,..,X_4) = \sum_{D_1,D_2} q(Y|D_1,D_2) \sum_{C_1,..,C_4} \prod_{i=1}^{2} q(D_i|C_{2i-1},C_{2i}) \prod_{j=1}^{4} q(C_j|X_j). \tag{5.6}$$

And the corresponding randomized prediction strategy is $\mathcal{Q} = \left\{\{q(C_i|X_i)\}_{i=1}^{4}, \{q(D_i|C_{2i-1},C_{2i}\}_{i=1}^{2}, q(Y|D_1,D_2)\right\}$. In this case the hypothesis space is the space of all hard partitions of X_i-s to C_i-s and of the pairs $\langle C_{2i-1}, C_{2i}\rangle$ to D_i-s. By repeating the analysis in theorem 4.1 we obtain that with a probability greater than $1-\delta$:

$$D_b(\hat{L}(\mathcal{Q}\|L(\mathcal{Q})) < \frac{F_1 + F_2 + |D_1||D_2|\ln|Y| + \frac{1}{2}\ln(4N) - \ln\delta}{N}, \tag{5.7}$$

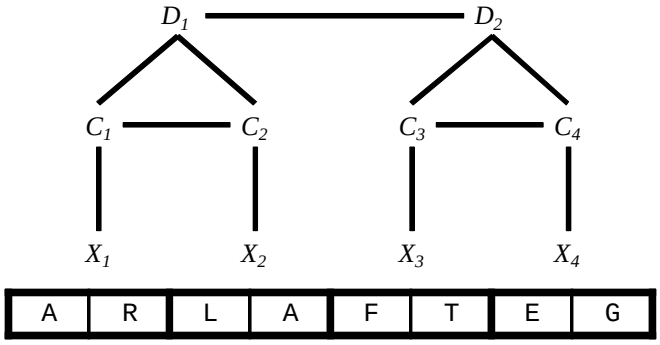


Figure 5.2: **Illustration of an application of models in Figure 5.1 to sequence modeling.** The sequence below is an imaginary subsequence of length 8 of a protein sequence. Each X_i corresponds to a pair of amino acids in the subsequence.

where

$$F_1 = \sum_{i=1}^{4} \left(n_i \tilde{I}(X_i; C_i) + m_i \ln n_i \right), \tag{5.8}$$

$$F_2 = \sum_{i=1}^{2} \left((m_{2i-1} m_{2i}) \tilde{I}(D_i; \langle C_{2i-1}, C_{2i} \rangle) + |D_i| \ln(m_{2i-1} m_{2i}) \right). \tag{5.9}$$

Observe that the $M \ln |Y|$ term in (4.6), which corresponds to the clique $\langle C_1, C_2, C_3, C_4, Y \rangle$, is replaced in (5.7) with terms which correspond to much smaller cliques $\langle C_1, C_2, D_1 \rangle$, $\langle C_3, C_4, D_2 \rangle$, and $\langle D_1, D_2, Y \rangle$. This factorization makes it possible to control the complexity of the partition and the tightness of the bound.

We provide an illustration of a possible application of the models in Figure 5.1. Imagine we intend to analyze protein sequences. Protein sequences are sequences over the alphabet of 20 amino acids. Subsequences of length 8 can reach $20^8 = 256 \cdot 10^8$ instantiations. Instead of studying this space directly, which would require an order of $256 \cdot 10^8$ samples, we can associate each X_i with a pair of amino acids - see Figure 5.2. The subspace of pairs of amino acids is only $20^2 = 400$ instances large and local interactions between adjacent pairs of amino acids can easily be studied. We can

cluster the pairs of amino acids into, let's say, 20 clusters C. Interactions between adjacent pairs of C-s in such a construction correspond to interactions between quadruples of amino acids. The subspace of quadruples is $20^4 = 16 \cdot 10^4$ instances large. However, the reduced subspace of pairs of C_i-s is only $20^2 = 400$ instances large. Thus, we have doubled the range of interactions, but remained at the same level of complexity. We can further cluster pairs of C_i-s (which correspond to quadruples of amino acids) into D_i-s and study the space of 8-tuples of amino acids while remaining at the same level of complexity.

The above approach shares the same basic principle already discussed in, for example, the collaborative filtering task. By clustering together similar pairs (and then quadruples) of amino acids we increase the statistical reliability of the observations, but reduce the resolution at which we process the data. The bound (5.7) suggests how this tradeoff between model resolution and statistical reliability can be optimized. It is further possible to derive analogs to (4.32) and (4.41) that apply to density estimation hierarchies as in Figure 5.1.b in a similar manner.

5.3 PAC-Bayesian Analysis of Graphical Models

The result in the previous section suggests a new approach to learning graphical models by providing a way to evaluate the expected performance of a graphical model on new data. Thus, instead of constructing a graphical model that fits the observed data it serves to construct a model with good generalization properties. Note that the prediction rule (5.6) and bound (5.7) both correspond to the undirected graph in Figure 5.1.a and to the directed graph in Figure 5.3. (In fact, Figure 5.1.a is a moralized counterpart of the directed acyclic graph in Figure 5.3 [28].)

The analysis used to derive bound (5.7) can be applied to any directed graphical model in the form of a tree (directed up, as in Figure 5.3) or its moralized counterpart. The analysis shows that the generalization power of these graphical models is determined by a tradeoff between empirical performance and the amount of information that is propagated up the tree.

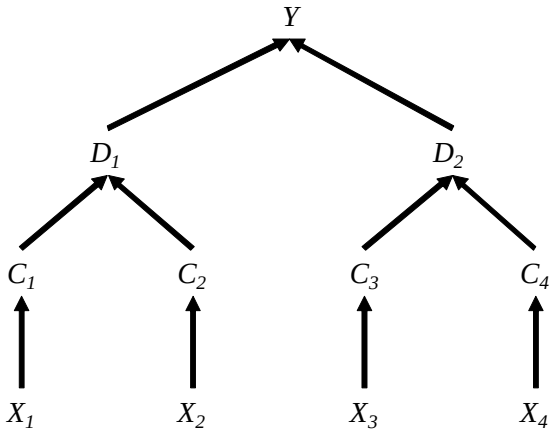


Figure 5.3: **Illustration of a directed graphical model.** A corresponding moral graph is depicted in Figure 5.1.a.

It is important to note that the PAC-Bayesian bound is able to utilize the factor form of distribution (5.6) and that bound (5.7) depends on the sizes of the tree cliques, but not on the size of the parameter space $\mathcal{X}_1 \times .. \times \mathcal{X}_4$. Further, a prior can be added over all possible directed graphs under consideration to obtain a PAC-Bayesian bound that will hold for all of them simultaneously. Development of efficient algorithms for optimization of the tree structure and extension of the results to more general graphical models will be key directions for future research.

Part III

Algorithms

Chapter 6

Bound Minimization Algorithms

In Chapter 4 we presented generalization bounds for discriminative prediction and density estimation with co-clustering. The bounds presented in theorems 4.1 and 4.4 hold for any prediction rule $\mathcal{Q}$ based on grid clustering of the parameter space $\mathcal{X}_1 \times .. \times \mathcal{X}_d$. In this chapter we address the question of how to find local optima of the bounds. Since the bounds are not convex in all of the parameters simultaneously it may be exponentially hard to find a globally optimal solution.

As we show in the applications section, the bounds are remarkably tight; however for practical purposes the tightness of the bounds may still be insufficient. In this chapter we suggest how to replace the bounds with a tradeoff that can be further fine tuned, e.g., via cross-validation, to improve their usability in practice.

Finally, we suggest an optimization procedure for finding the global optimum of the bound for discriminative prediction based on a single parameter suggested in chapter 5.

6.1 Minimization of the PAC-Bayesian Bound for Discriminative Prediction with Grid Clustering

We start with minimization of the PAC-Bayesian bound for discriminative prediction based on grid clustering (4.6) suggested in theorem 4.1. For convenience we quote the bound (4.6) below once again:

$$D_b(\hat{L}(\mathcal{Q})\|L(\mathcal{Q})) < \frac{\sum_{i=1}^{d}\left(n_i\tilde{I}(X_i;C_i)+m_i\ln n_i\right)+\left(\prod_{i=1}^{d}m_i\right)\ln|Y|+K}{N},$$

where $K=\frac{1}{2}\ln(4N)-\ln\delta$. We further rewrite it in a slightly different way:

$$D_b(\hat{L}(\mathcal{Q})\|L(\mathcal{Q})) < \frac{\sum_{i=1}^{d} n_i\tilde{I}(X_i;C_i)+K'}{N}, \tag{6.1}$$

where

$$K'=\sum_{i=1}^{d} m_i\ln n_i+\left(\prod_{i=1}^{d}m_i\right)\ln|Y|+\frac{1}{2}\ln(4N)-\ln\delta.$$

Note that K' depends on the number of clusters m_i used along each dimension, but not on a specific form of a grid partition. Once the number of clusters used along each dimension has been chosen, K' is constant.

The minimization problem corresponding to (6.1) can be stated as follows:

$$\min_{\mathcal{Q}} \; L \qquad s.t. \quad D_b(\hat{L}(\mathcal{Q})\|L)=\frac{\sum_{i=1}^{d} n_i\tilde{I}(X_i;C_i)+K'}{N}. \tag{6.2}$$

It is generally possible to solve the minimization problem (6.2) directly using alternating projection methods - see, e.g., [72] for such an approach to solving a similar minimization problem for linear classifiers. We choose a slightly different way that further enables us to compensate for the imperfection of the bounds. Since K' is constant, $L(\mathcal{Q})$ depends on the tradeoff

between $\hat{L}(\mathcal{Q})$ and $\sum_{i=1}^{d} n_i \tilde{I}(X_i; C_i)$, which can be written as follows:

$$\mathcal{F}(\mathcal{Q}) = \beta N \hat{L}(\mathcal{Q}) + \sum_{i=1}^{d} n_i \tilde{I}(X_i; C_i). \tag{6.3}$$

The minimization problem (6.2) is then replaced by:

$$\min_{\mathcal{Q}} \beta N \hat{L}(\mathcal{Q}) + \sum_{i=1}^{d} n_i \tilde{I}(X_i; C_i). \tag{6.4}$$

In general, every value of β yields a different solution to the minimization problem (6.4). The optimum of (6.2) (which is computationally hard to find) corresponds to some specific value of β. Hence, by scanning the possible values of β and minimizing (6.4) it is virtually possible to find the optimum of (6.2) (only virtually, because finding the global optimum of (6.4) is computationally hard as well). However, the tradeoff (6.3) provides us an additional degree of freedom. In cases where the bound (4.6) is not sufficiently tight for practical applications it is possible to tune the tradeoff by determining the desired value of β via cross-validation instead of back-substitution into the bound.

The minimization problem (6.4) is closely related to the rate distortion tradeoff in information theory [27]. Since $\hat{L}(\mathcal{Q})$ is linear in $\mathcal{Q}$ and $\tilde{I}(X_i; C_i)$ is convex in $\mathcal{Q}$, for $d = 1$ it is possible to find the global minimum of $\mathcal{F}(\mathcal{Q})$. However, for $d \geq 2$ only a local minimum can be achieved. To find a local minimum of $\mathcal{F}(\mathcal{Q})$ we adapt an iterative minimization EM-like alternating projection procedure, very similar to the Blahut-Arimoto algorithm for minimization of the rate distortion function [5, 17, 27]. For the sake of simplicity of the notations we restrict ourselves to the case of $d = 2$.

The Lagrangian corresponding to the minimization problem (6.4) is:

$$\mathcal{L}(\mathcal{Q}) = \beta N \hat{L}(\mathcal{Q}) + \sum_{i=1}^{2} n_i \tilde{I}(X_i; C_i) + \sum_{i=1}^{2} \sum_{x_i \in \mathcal{X}_i} \nu(x_i) \sum_{c_i} q(c_i|x_i)$$

$$+ \sum_{c_1,c_2} \nu(c_1,c_2) \sum_{y} q(y|c_1,c_2), \tag{6.5}$$

where ν-s are Lagrange multipliers corresponding to normalization constraints on $\{q(C_i|X_i)\}_{i=1}^2$ and $q(Y|C_1,C_2)$. In order to minimize $\mathcal{L}(\mathcal{Q})$ we write $\hat{L}(\mathcal{Q})$ explicitly:

$$\begin{aligned}
\hat{L}(\mathcal{Q}) &= \sum_{x_1,x_2,y} \hat{p}(x_1,x_2,y) \sum_{y'} q(y'|x_1,x_2) l(y,y') \\
&= \sum_{x_1,x_2,y} \hat{p}(x_1,x_2,y) \sum_{y',c_1,c_2} q(y'|c_1,c_2) q(c_1|x_1) q(c_2|x_2) l(y,y') \\
&= \sum_{y,y'} l(y,y') \sum_{c_1,c_2} q(y'|c_1,c_2) \sum_{x_1,x_2} q(c_1|x_1) \hat{p}(x_1,x_2,y) q(c_2|x_2).
\end{aligned}$$

We further derive $\hat{L}(\mathcal{Q})$ with respect to $q(C_1|X_1)$. The derivative with respect to $q(C_2|X_2)$ is similar. To improve the readability of the following equations we use lower case letters to denote variables that change in summations and capital letters to denote variables that are fixed in summations.

$$\frac{\partial \hat{L}(\mathcal{Q})}{\partial q(C_1|X_1)} = \sum_{y,y'} l(y,y') \sum_{x_2,c_2} \hat{q}(y'|C_1,c_2) p(X_1,x_2,y) q(c_2|x_2). \tag{6.6}$$

Recall that:

$$\tilde{I}(X_i;C_i) = \frac{1}{n_i} \sum_{x_i,c_i} q(c_i|x_i) \ln \frac{q(c_i|x_i)}{\tilde{q}(c_i)}$$

and

$$\tilde{q}(c_i) = \frac{1}{n_i} \sum_{x_i} q(c_i|x_i).$$

Hence:

$$\frac{\partial n_i \tilde{I}(X_i;C_i)}{\partial q(C_i|X_i)} = \ln \frac{q(C_i|X_i)}{\tilde{q}(C_i)}.$$

Derivatives of the remaining terms in $\mathcal{L}(\mathcal{Q})$ provide normalization for the corresponding variables. Thus, taking the derivative of $\mathcal{L}(\mathcal{Q})$ with respect to $q(C_i|X_i)$, equating it to zero and reorganizing the terms we obtain a set

of self-consistent equations that can be iterated until convergence:

$$\tilde{q}_t(c_i) = \frac{1}{n_i} \sum_{x_i} q_t(c_i|x_i) \tag{6.7}$$

$$q_{t+1}(c_i|x_i) = \frac{\tilde{q}_t(c_i)}{Z_{t+1}(x_i)} e^{-\beta N \frac{\partial \hat{L}(\mathcal{Q}_t)}{\partial q(c_i|x_i)}} \tag{6.8}$$

$$Z_{t+1}(x_i) = \sum_{c_i} q_{t+1}(c_i|x_i) \tag{6.9}$$

$$q_{t+1}(y|c_1, c_2) = \delta[y, y^*_{t+1}(c_1, c_2)] \tag{6.10}$$

$$y^*_{t+1}(c_1, c_2) = \arg\min_{y'} \sum_{y} l(y, y') \sum_{x_1, x_2} q_{t+1}(c_1|x_1) \hat{p}(x_1, x_2, y) q_{t+1}(c_2|x_2), \tag{6.11}$$

where $\delta[\cdot, \cdot]$ is the Kronecker delta function, $\frac{\partial \hat{L}(\mathcal{Q})}{\partial q(C_i|X_i)}$ is given by (6.6), and the subindex t denotes the iteration number. Equations (6.10) and (6.11) correspond to minimization of $\hat{L}(\mathcal{Q})$ with respect to $q(Y|C_1, C_2)$ and generally depend on the loss function. For the zero-one loss $y^*(c_1, c_2)$ is the most frequent value of y appearing in the $\langle c_1, c_2 \rangle$ partition cell, for the absolute loss it is the median value, for the quadratic loss it is the average value. We summarize the algorithm in Figure 6.1. We note that the quadratic loss $y^*(c_1, c_2)$, which is the average value in this case, can fall out of the finite space of labels $\mathcal{Y}$ and generally a separate analysis is required for this case (which is beyond the scope of this work). However, in practice the algorithm can still be applied.

6.2 Minimization of the PAC-Bayesian Bound for Density Estimation

Similar to the PAC-Bayesian bound for discriminative prediction, the PAC-Bayesian bound for density estimation (4.41) depends on the tradeoff:

$$\mathcal{G}(\mathcal{Q}) = -\beta N I(\hat{p}_{\mathcal{Q}}(C_1, C_2)) + \sum_{i=1}^{2} n_i \tilde{I}(X_i; C_i). \tag{6.12}$$

Algorithm 6.1 Algorithm for alternating projection minimization of $\mathcal{F}(\mathcal{Q}) = \beta N \hat{L}(\mathcal{Q}) + \sum_{i=1}^{2} n_i \tilde{I}(X_i; C_i)$.

Input: $\hat{p}(x_1, x_2, y)$, N, n_1, n_2, m_1, m_2, $l(y, y')$, $|Y|$, β.
Initialize: $q_0(C_i|X_i)$ and $q_0(Y|C_1, C_2)$ randomly.
$\tilde{q}_0(c_i) \leftarrow \frac{1}{n_i} \sum_{x_i} q_t(c_i|x_i)$
repeat
  for $i = 1, 2$ **do**
    $q_{t+1}(c_i|x_i) \leftarrow \tilde{q}_t(c_i) e^{-\beta N \frac{\partial \hat{L}(\mathcal{Q}_t)}{\partial q(c_i|x_i)}}$
    $Z_{t+1}(x_i) \leftarrow \sum_{c_i} q_{t+1}(c_i|x_i)$
    $q_{t+1}(c_i|x_i) \leftarrow \frac{q_{t+1}(c_i|x_i)}{Z_{t+1}(x_i)}$
    $\tilde{q}_{t+1}(c_i) \leftarrow \frac{1}{n_i} \sum_{x_i} q_{t+1}(c_i|x_i)$
    $y^*_{t+1}(c_1, c_2) \leftarrow \arg\min_{y'} \sum_y l(y, y') \sum_{x_1, x_2} q_{t+1}(c_1|x_1) \hat{p}(x_1, x_2, y) q_{t+1}(c_2|x_2)$
    $q_{t+1}(y|c_1, c_2) \leftarrow \delta[y, y^*_{t+1}(c_1, c_2)]$
    $t \leftarrow t + 1$
  end for
until convergence
return $\{q(C_i|X_i)\}_{i=1}^2, q(Y|C_1, C_2)$ from the last iteration.

All other terms in (4.41) do not depend on the specific form of grid partition $\mathcal{Q}$. (As in the previous section we restrict ourselves to $d = 2$.) Unfortunately, $-I(\hat{p}_{\mathcal{Q}}(C_1, C_2))$ is concave in $q(C_i|X_i)$-s, whereas $\tilde{I}(X_i; C_i)$ is convex in $q(C_i|X_i)$. Therefore, alternating projection methods are hard to apply. Instead, $\mathcal{G}(\mathcal{Q})$ can be minimized using sequential minimization [91, 32]. The essence of sequential minimization method is that we start with some random assignment $q(C_i|X_i)$ and then iteratively take x_i-s out of their clusters and reassign them to new clusters, so that $\mathcal{G}(\mathcal{Q})$ is minimized. This approach returns a hard partition of the data (i.e., each x_i is deterministically assigned to a single c_i). The algorithm is given in Figure 6.2.

Algorithm 6.2 Algorithm for sequential minimization of $\mathcal{G}(\mathcal{Q}) = -\beta N I(\hat{p}_{\mathcal{Q}}(C_1, C_2)) + \sum_{i=1}^{2} n_i \tilde{I}(X_i; C_i)$.

Input: $\hat{p}(x_1, x_2)$, N, n_1, n_2, m_1, m_2, β.
Initialize: $q_0(C_i|X_i)$ randomly.
repeat
  for all $x_1 \in \mathcal{X}_1$ and all $x_2 \in \mathcal{X}_2$ according to some random order over $\mathcal{X}_1$ and $\mathcal{X}_2$: **do**
    for $i = 1, 2$ **do**
      Select $x_i \in \mathcal{X}_i$ according to the order selected above.
      Compute $\mathcal{G}(\mathcal{Q})$ for each possible assignment of x_i to $c_i \in \{1, .., m_i\}$

      Reassign x_i to c_i such that $\mathcal{G}(\mathcal{Q})$ is minimized.
      Update $\hat{p}_{\mathcal{Q}}(C_1, C_2) \leftarrow \sum_{x_1, x_2} q(C_1|x_1)\hat{p}(x_1, x_2)q(C_2|x_2)$.
    end for
  end for
until no reassignments further minimize $\mathcal{G}(\mathcal{Q})$.
return $\{q(C_i|X_i)\}_{i=1}^{2}$ from the last iteration.

6.3 Minimization of the PAC-Bayesian Bound for Discriminative Prediction when $d = 1$

In discriminative prediction based on a single parameter ($d = 1$) the tradeoff (6.3) is simplified to:

$$\mathcal{F}(\mathcal{Q}) = \beta N \hat{L}(\mathcal{Q}) + n\tilde{I}(X; Y). \tag{6.13}$$

The algorithm 6.1 then simplifies to alternating minimization between two sets that are convex in $q(Y|X)$, exactly as in minimization of the rate distortion function [27]. Hence, the minimization achieves the global optimum of $\mathcal{F}(\mathcal{Q})$ in this case. Further, by a linear scan of the values of β it is possible to achieve the global optimum of (5.4).

The algorithm for minimization of (6.13) is given in Figure 6.3. To derive the algorithm we use the fact that

$$\hat{L}(\mathcal{Q}) = \sum_{x,y} \hat{p}(x)\hat{p}(y|x) \sum_{y'} q(y'|x) l(y, y')$$

and

$$\frac{d\hat{L}(\mathcal{Q})}{dq(Y|X)} = \hat{p}(X)\sum_{y'}\hat{p}(y'|X)l(Y,y').$$

Note that the derivative is in fact constant.

Algorithm 6.3 Algorithm for alternating projection minimization of $\mathcal{F}(\mathcal{Q}) = \beta N\hat{L}(\mathcal{Q}) + n\tilde{I}(X;Y)$.

Input: $\hat{p}(x,y)$, N, n, $l(y,y')$, $|Y|$, β.
Initialize: $q_0(Y|X)$ arbitrarily.
$\tilde{q}_0(y) \leftarrow \frac{1}{n}\sum_x q_t(y|x)$
repeat
  $q_{t+1}(y|x) \leftarrow \tilde{q}_t(y)e^{-\beta N\frac{d\hat{L}(\mathcal{Q})}{dq(y|x)}}$
  $Z_{t+1}(x) \leftarrow \sum_y q_{t+1}(y|x)$
  $q_{t+1}(y|x) \leftarrow \frac{q_{t+1}(y|x)}{Z_{t+1}(x)}$
  $\tilde{q}_{t+1}(y) \leftarrow \frac{1}{n}\sum_x q_{t+1}(y|x)$
  $t \leftarrow t+1$
until convergence
return $q(Y|X)$ from the last iteration.

Part IV

Applications

Chapter 7

Applications

In this chapter we demonstrate several illustrative applications of the bounds developed in this thesis.

7.1 Collaborative Filtering

The problem of collaborative filtering was discussed in the previous chapters. The goal of collaborative filtering is to complete the missing entries in a viewers by movies ratings matrix. This problem attracted a great deal of attention recently thanks to the Netflix challenge [1]. Since our goal here is mainly to illustrate our approach to co-clustering via the PAC-Bayesian bounds rather than to solve the large scale challenge we concentrate on a much smaller MovieLens 100K dataset [2]. The dataset consists of 100,000 ratings on a five-star scale for 1,682 movies by 943 users. We take the five non-overlapping splits of the dataset into an 80% train and a 20% test subsets provided at the MovieLens website. We stress that the training data are extremely sparse - only 5% of the training matrix entries are populated, whereas 95% of the values are missing.

To measure the accuracy of our algorithm we use mean absolute error (MAE) metrics, which is commonly used for evaluation on this dataset [43]. Let $\check{p}(x_1, x_2, y)$ be the distribution over $\langle X_1, X_2, Y \rangle$ in the test set. The

mean absolute error is defined as:

$$MAE = \sum_{x_1,x_2,y} \check{p}(x_1,x_2,y) \sum_{y'} q(y'|x_1,x_2)|y-y'|. \tag{7.1}$$

In previous work the best MAE reported for this dataset was 0.73 [43]. It is worth recalling that the ratings are on a five-star scale, thus a MAE of 0.73 means that on average the predicted rating is 0.73 stars (less than one star) far from the observed rating. The maximal possible error is 4 (which occurs if we predict one star instead of five or vice versa), which determines the scale on which all the results should be judged.

In [80] we improved the MAE on this dataset to 0.72 using the MDL formulation of co-clustering. In the MDL formulation the co-clustering solutions are evaluated by the total description length, which includes the length of the description of assignments of X_i-s to C_i-s together with the length of the description of the ratings given the assignments. For fixed numbers of clusters m_i-s used along each dimension, the MDL solution corresponds to optimization of the tradeoff (6.3) for $\beta = 1$. For convenience we cite (6.3) below once again:

$$\mathcal{F}(\mathcal{Q}) = \beta N \hat{L}(\mathcal{Q}) + \sum_{i=1}^{d} n_i \tilde{I}(X_i;C_i).$$

In the MDL formulation of co-clustering developed in [80] only hard (deterministic) assignments of X_i-s to C_i-s were considered. The best performance of 0.72 was achieved at $m_1 = 13$ and $m_2 = 6$ with beyond 1% sensitivity to small changes in m_1 and in m_2 both in the description length and in the prediction accuracy. The deviation in prediction accuracy between the five splits of the MovieLens dataset was below 0.01.

In this work we implemented algorithm 6.1 for minimization of $\mathcal{F}(\mathcal{Q})$ (for an arbitrary value of β) and applied it to the MovieLens dataset. There are four major differences between the algorithm 6.1 and the algorithm for minimization of $\mathcal{F}(\mathcal{Q})$ suggested in [80] that should be highlighted:

- The algorithm 6.1 considers soft assignments of X_i-s to C_i-s.

- The algorithm 6.1 is a gradient descent algorithm rather than the sequential optimization algorithm suggested in [80]. Note that this point is neither positive nor negative, since sequential optimization algorithms are very powerful and especially in hard cases can outperform gradient descent methods. The advantage of gradient descent methods is in their mathematical elegance, faster convergence (although in the hard cases it may be fast convergence to trivial, but strong attractors), and the ability to handle soft assignments.

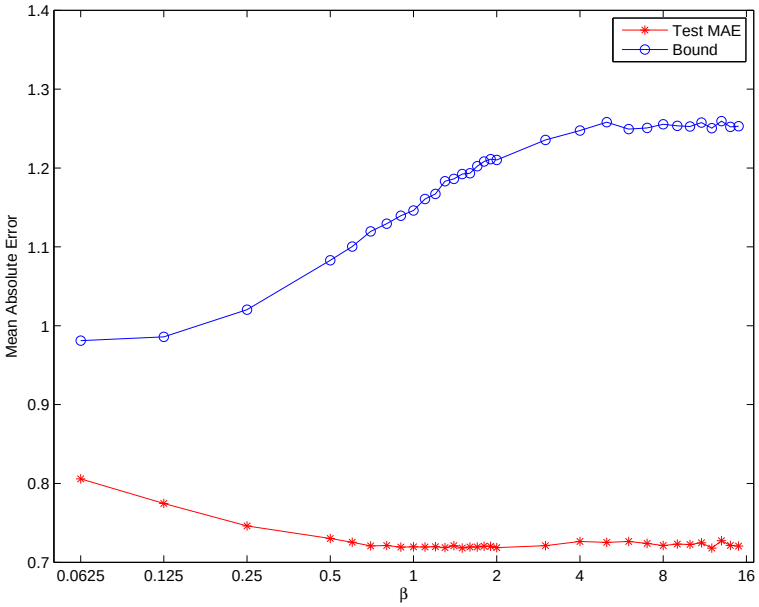
- Algorithm 6.1 directly optimizes a given loss function (MAE in the case of MovieLens) rather than the description length, which is only indirectly related to the loss function.

- Algorithm 6.1 considers arbitrary values of β. (However, the algorithm in [80] can be easily extended to handle arbitrary values of β.) As we will show below, $\beta = 1$ is not always optimal.

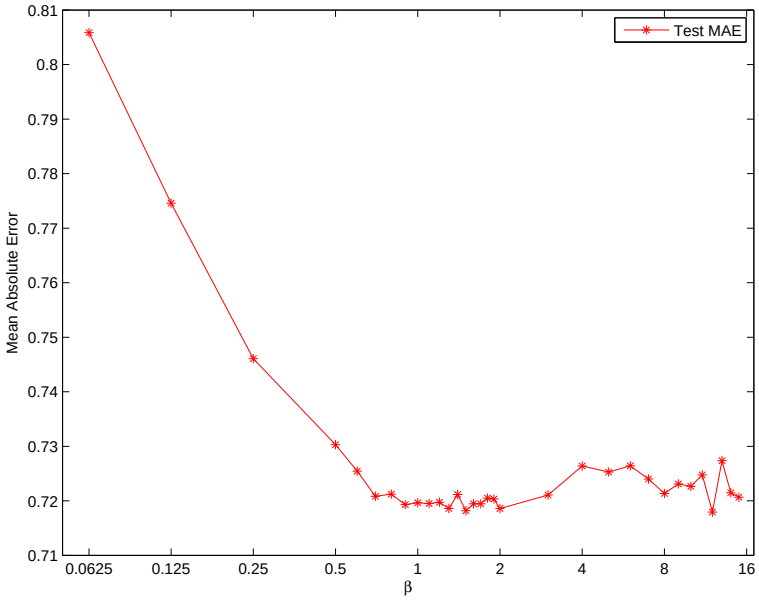
We conducted three experiments with algorithm 6.1. In all three experiments we fix the numbers of clusters m_1 and m_2 used along both dimensions and analyze the MAE loss on the test set and the value of the bound (4.6) as a function of β. In each experiment, for each of the five splits of the dataset into train and test sets mentioned earlier, and for each value of β we apply 10 random initializations of the algorithm. The solution $\mathcal{Q}$ corresponding to the best value of $\mathcal{F}(\mathcal{Q})$ per each data split and per each value of β is then selected. We further calculate the average of the results over the dataset splits to produce the graphs of the bound values and test MAE as functions of β.

In the first experiment we verify that we are able to reproduce the results achieved previously in [80]. We set $m_1 = 13$ and $m_2 = 6$, as the best values obtained in [80] and apply algorithm 6.1. The results are presented in Figure 7.1. We make the following observations based on this experiment:

- The performance of algorithm 6.1 is comparable to the performance achieved in [80] with sequential optimization.



(a) Bound (4.6).



(b) Test Loss (zoom into subfigure a.).

Figure 7.1: **Co-clustering of the MovieLens dataset into 13x6 clusters.** Figure a. shows the value of bound (4.6) together with the MAE on the test set as a function of β. Figure b. zooms into MAE on the test set. The values of β are on a log scale. See text for further details.

- The optimal performance is achieved at β close to one, which corresponds to the MDL functional optimized in [80].

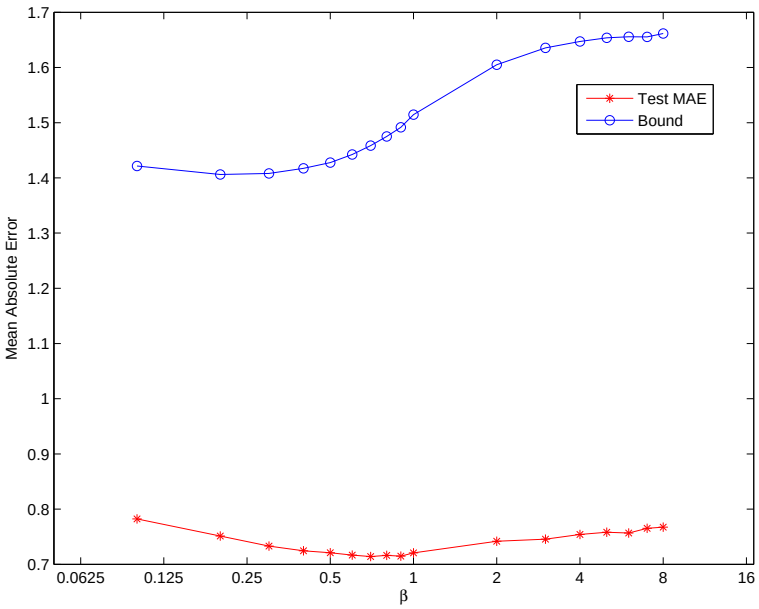
- The values of the bound are meaningful (recall that the maximal possible loss is 4; thus the bound value of ~ 1.25 is informative).

- The bound is 25%-75% far from the test error.

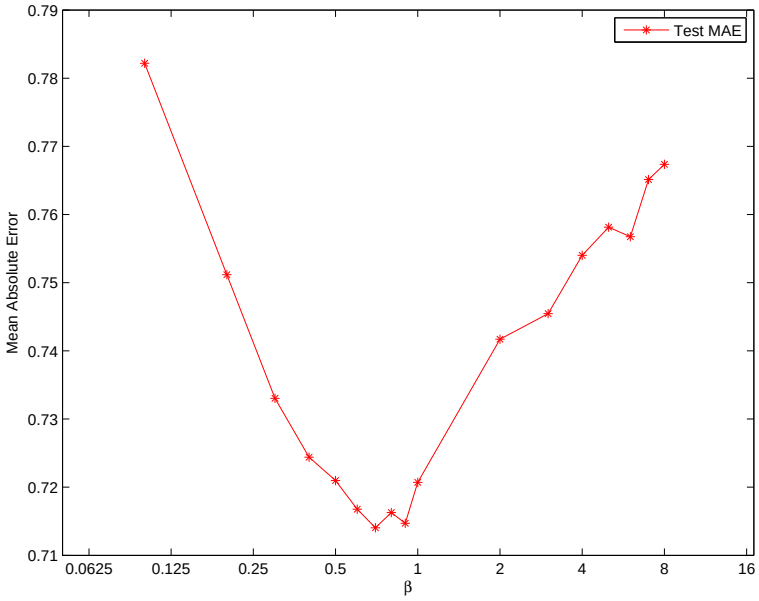
- The bound does not follow the shape of the test loss. According to the bound in this task it is best to assign all the data to one big cluster. This is partially explained by the fact that this is a hard problem and the improvement in the empirical loss $\hat{L}(\mathcal{Q})$ achieved by co-clustering is relatively small. For the best co-clustering solution found $\hat{L}(\mathcal{Q}) \approx 0.67$, whereas if we assign all the data to one big cluster $\hat{L}(\mathcal{Q}) \approx 0.89$. Thus, the improvement in $\hat{L}(\mathcal{Q})$ achieved by the clustering is only about 30% when the tightness of the bound is 25%-75%. This is clearly insufficient to apply the bound as the main guideline for model order selection. However, it is possible to set the value of β in the tradeoff $\mathcal{F}(\mathcal{Q})$ via cross-validation and obtain remarkably good results. It should be pointed out that the tradeoff $\mathcal{F}(\mathcal{Q})$ was derived from the bound, thus even though the analysis is not perfectly tight it produced a useful practical result.

- Note that in the setting of this experiment the small values of m_1 and m_2 provide "natural regularization"; thus there is no significant decrease in performance when we increase β beyond 1. This will change in the following experiments.

One of the major advantages of bound (4.6) and the tradeoff $\mathcal{F}(\mathcal{Q})$ derived from it is that it mainly penalizes the effective complexity of the solution rather than the gross number of clusters used. The practical implication of this property is that we can initialize the optimization algorithm with more clusters than are actually required to solve a problem and the algorithm will automatically adjust the extent to which it uses the available clusters. This property is verified in the following two experiments. In the



(a) Bound (4.6).



(b) Test Loss (zoom into subfigure a.).

Figure 7.2: **Co-clustering of the MovieLens dataset into 50x50 clusters.** Figure a. shows the value of bound (4.6) together with the MAE on the test set as a function of β. Figure b. zooms into MAE on the test set. The values of β are on a log scale. See text for further details.

first of these we initialize algorithm 6.1 with $m_1 = m_2 = 50$ clusters along each dimension. The result of optimization of $\mathcal{F}(\mathcal{Q})$ as a function of β is presented in Figure 7.2. We make the following observations based on this experiment:

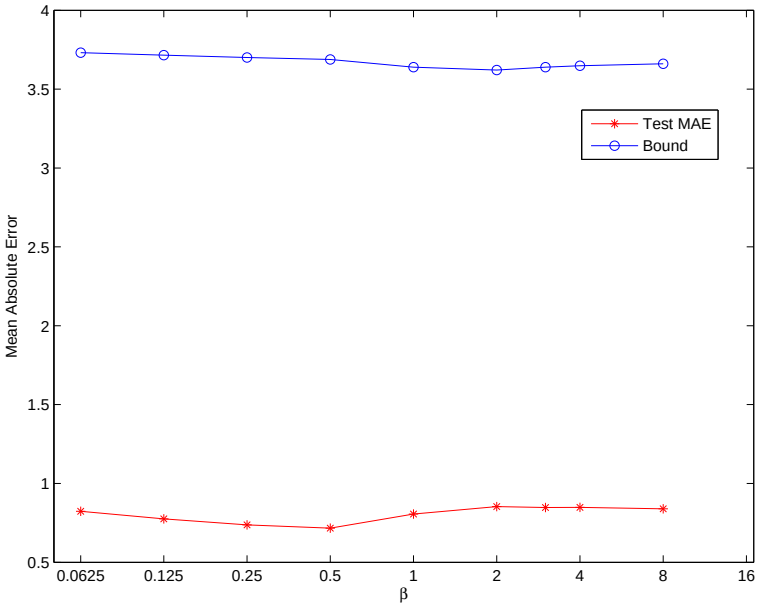
- The best performance of the test MAE (0.72) achieved in the previous setting with $m_1 = 13$ and $m_2 = 6$ is achieved in the new setting with $m_1 = m_2 = 50$ as well. This supports the ability of the algorithm to operate with more clusters than are actually required by the problem and to adjust the complexity of the solution automatically.

- Note that the optimal value of β in this setting is below 1. In particular, this implies that the MDL formulation, which corresponds to $\beta = 1$ would overfit in this case. The role of the regularization parameter β is also more evidently expressed here compared to the preceding experiment.

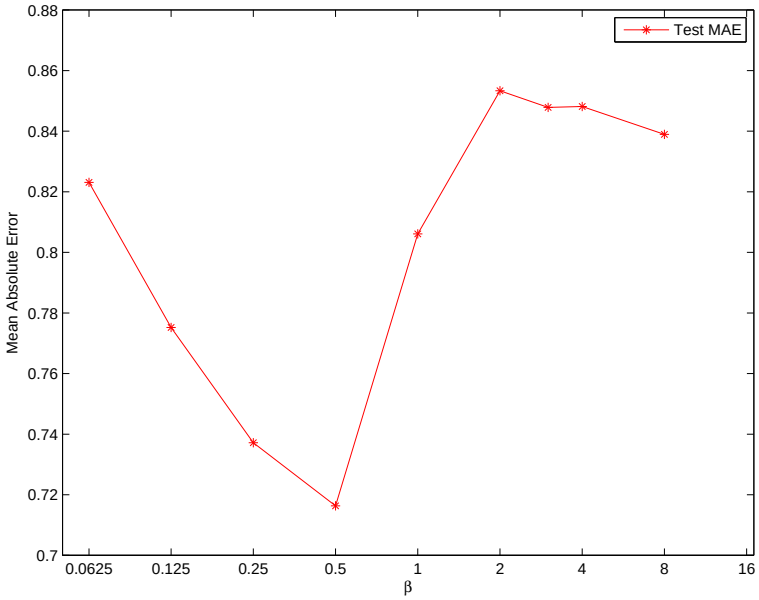
- The values of the bound, although less tight than in the previous case, are still meaningful. The shape of the bound becomes closer to the shape of the test loss, although in light of the preceding experiment we would not attribute importance to it and still prefer to set the value of β via cross-validation.

In our last experiment we went to the extreme case of taking $m_1 = m_2 = 283$. Note that the size of the cluster space $m_1 m_2$ in this case is $m_1 m_2 = 80,089$ and is equal to the size of the train set, $N = 80,000$. The implication is that extensive use of all available clusters can result in a situation where each partition cell contains an order of a single observation, which is clearly insufficient for statistically reliable predictions. Thus, in this experiment the number of clusters provides no regularization at all and the only parameter responsible for regularization of the model is the tradeoff parameter β. The result of the experiment is presented in Figure 7.3. We highlight the following points regarding this experiment:

- The best performance of the test MAE (0.72) is achieved in this experiment as well. This further stresses the ability to have full control



(a) Bound (4.6).



(b) Test Loss (zoom into subfigure a.).

Figure 7.3: **Co-clustering of the MovieLens dataset into 283x283 clusters.** Figure a. shows the value of bound (4.6) together with the MAE on the test set as a function of β. Figure b. zooms into MAE on the test set. The values of β are on a log scale. See text for further details.

over regularization of the model via parameter β of the tradeoff $\mathcal{F}(\mathcal{Q})$.

- The role of regularization parameter β is further increased in this experiment compared to the previous two. The optimal value of β here is clearly below 1 (the optimal $\beta \approx 0.5$), suggesting that the MDL solution would be overfitting.

- The value of the bound still remains meaningful, although it is already quite far from the test error. The shape of the bound does not seem to provide useful information and the value of β should be set via cross-validation.

7.2 Prediction by a Single Parameter ($d = 1$) and Feature Rating

In this section we provide a series of applications of the PAC-Bayesian bound for classification by a single feature (5.4). We use algorithm 6.3 to find the optimal prediction rule $q^*(y|x)$ that minimizes the bound. More precisely, we apply the algorithm to minimize the tradeoff $\mathcal{F}(\mathcal{Q})$ given in (6.13) for a set of values of β and select the value of β for which the value of the bound (5.4) is minimal. Figure 7.4 illustrates that the bound (5.4) has a single global optimum as a function of β. Hence, the above procedure achieves the global optimum of (5.4).

We further compare the PAC-Bayesian bound with Occam's razor bound (3.24) applied to the maximum likelihood classification rule. For the zero-one loss the maximum likelihood classification rule $q_{ml}(X)$ returns for each value of x the most frequent value of Y that appeared with that x in the sample: $q_{ml}(x) = \arg\max_y \hat{p}(y|x)$.

The results presented here generally follow the experiments reported previously in [82]. However, there are two improvements compared to the previous work that should be mentioned. First, we eliminate the $\frac{1}{4}(\ln(n) + 1)^2$ factor from the bounds, and second, we directly optimize the bounds in their KL-divergence form rather than the square root approximation as was

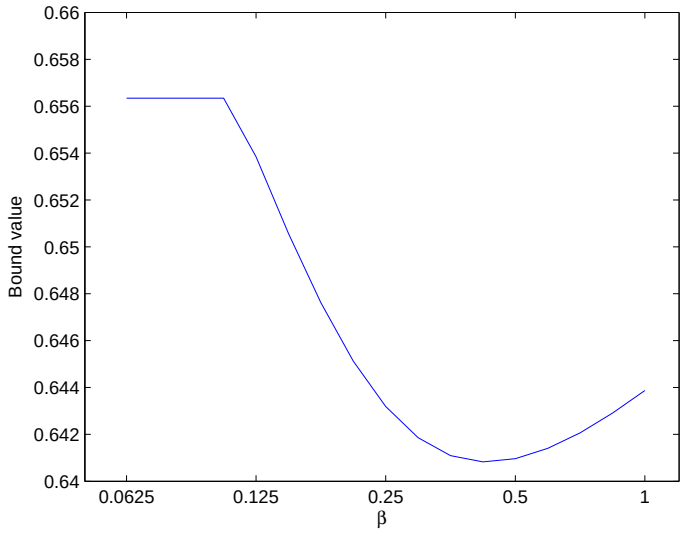


Figure 7.4: **The value of PAC-Bayesian bound** (5.4) **as a function of** β **in the** $\mathcal{F}(\mathcal{Q})$ **tradeoff.** In this experiment we selected a range of values of β and optimized $\mathcal{F}(\mathcal{Q})$ for each value of β using algorithm 6.3 for the first parameter in the CMC dataset. Recall that the algorithm 6.3 achieves the global optimum of $\mathcal{F}(\mathcal{Q})$. For each value of β we calculated the value of bound (5.4) corresponding to the solution that optimizes $\mathcal{F}(\mathcal{Q})$ and plotted it in this figure. From the figure it is clear that the global optimum of (5.4) can easily be found. The shape of the dependence is typical for all other features and experiments.

done previously in [82]. These two factors (mainly the first one) lead to a slight improvement in the results.

The experiments were conducted on four datasets obtained from the UCI Machine Learning Repository: Contraceptive Method Choice (CMC), Mushrooms, Letters, and Nursery. In all the experiments we used 5 random partitions of the data into 80% train and 20% test subsets. Table 7.1 provides a short summary of the main parameters of the datasets. See [6] for a full description.

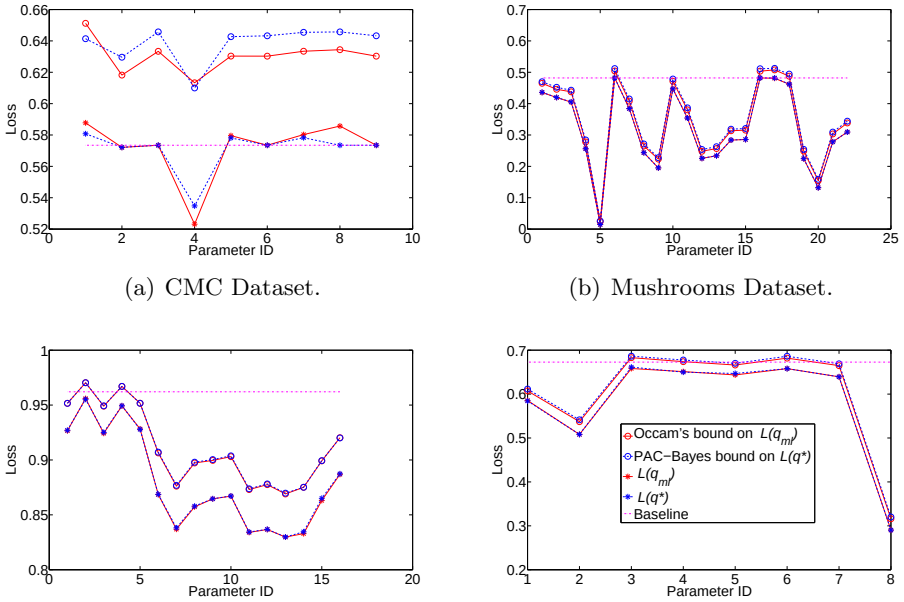
In Figure 7.5 we present the test loss of the maximum likelihood prediction rule $q_{ml}(x)$ and the test loss of the classification rule $q^*(y|x)$ that

Table 7.1: **Description of the datasets:** for every dataset we indicate the number of features, d, a list of cardinalities of the features, n_i, the number of labels, $|\mathcal{Y}|$, and a train set size, N, which is 80% of each dataset size.

DATA SET	d	n_i-S	$\lvert\mathcal{Y}\rvert$	N
CMC	9	34, 4, 4, 15, 2, 2, 4, 4, 2	3	1,178
MUSHROOMS	22	6, 4, 10, 2, 9, 2, 2, 2, 12, 2, 5, 4, 4, 9, 9, 1, 4, 3, 5, 9, 6, 7, 2	2	6,499
LETTERS	16	16 FOR ALL n_i-S	26	16,000
NURSERY	8	3, 5, 4, 4, 3, 2, 3, 3	5	10,368

minimizes the bound (5.4). Both prediction rules perform similarly. The same figure presents the values of PAC-Bayesian bound (5.4) for $q^*(y|x)$ and the values of Occam's razor bound (3.24) for $q_{ml}(x)$. The performance of both bounds is also similar. Note that the bounds are exceptionally tight and that the bounds are less than 15% away from the test error in all cases.

We conclude this section by comparing bounds (3.24) and (5.4) applied to feature ranking with the standard empirical mutual information $\hat{I}(X;Y) = \sum_{x,y} \hat{p}(x)\hat{p}(y|x) \ln \frac{\hat{p}(y|x)}{\hat{p}(y)}$ and the normalized correlation coefficient $Corr(X;Y) = \frac{Cov(X,Y)}{\sqrt{Var(X)Var(Y)}}$ indices. We compare the agreement between the Top-1, Top-2, and Top-3 parameter subsets suggested by the indices with the corresponding test-based sets in Figure 7.6. For the Top-1 choice (the best single parameter) bounds (3.24) and (5.4) are clearly superior to mutual information and normalized correlation in that they provide a significant level of success in two cases where the other two indices completely fail. For the Top-2 choice there is a slight advantage over the mutual information and a clear advantage over the normalized correlation, which fails completely on the —Mushrooms dataset. In Top-3 the bounds perform similarly to the mutual information and are still superior to the normalized correlation. The performance of Occam's razor and the PAC-Bayesian bound is comparable.



(a) CMC Dataset.

(b) Mushrooms Dataset.

(c) Letters Dataset.

(d) Nursery Dataset

Figure 7.5: **Application of bounds** (5.4) **and** (3.24)**.** This figure displays an application of PAC-Bayesian bound (5.4) and Occam's razor bound (3.24) to the four datasets discussed in text. The legend in subfigure d. corresponds to all the graphs. The graphs contain the test loss $L(q_{ml})$ and the value of Occam's razor bound (3.24) for the maximum likelihood prediction rule $q_{ml}(x) = \arg\max_y \hat{p}(y|x)$. They also depict the test loss $L(q^*)$ and the value of the PAC-Bayesian bound (5.4) for the prediction rule $q^*(Y|X)$ that minimizes it. Each point on the graphs is an average over 5 random splits of the corresponding dataset. Baseline corresponds to the performance level that can be achieved by predicting the test labels using a marginal distribution of Y on the train set. All the calculations are done per parameter; i.e., each point on the graphs corresponds to a separate prediction rule based on the corresponding parameter. For better visibility of the points they have been connected with lines, but the lines have no meaning.

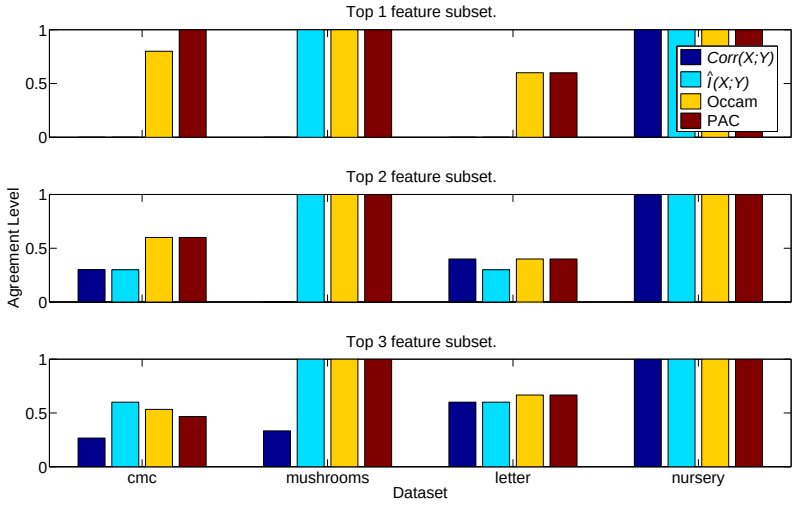


Figure 7.6: **Feature Ranking.** Agreement of $Corr(X;Y)$, $\hat{I}(X;Y)$, and Occam's razor bound (3.24) and the PAC-Bayesian bound (5.4) with the test set on the Top-1, Top-2, and Top-3 feature subsets.

Chapter 8

Discussion and Future Work

8.1 Discussion

There are two major messages to be drawn from this thesis. While they could have been lost in the technical details of the preceding chapters this discussion section provides an opportunity to highlight them once again.

The primary message is that we do not learn structure just for the sake of learning structure, but rather the structure we have learned facilitates the solution of a higher level goal. Thus it is critical to identify and state this high level goal clearly and evaluate the quality of the structures we have learned in terms of their utility in this context.

The second message is that the PAC-Bayesian bounds are a handy tool for theoretical analysis of the expected performance of structure-based learning methods. Hypothesis spaces based on structured models usually exhibit natural heterogeneity, because structures can be differentiated by their complexity. PAC-Bayesian bounds are a useful way to utilize this heterogeneity and to derive tight bounds that depend on the tradeoff between model complexity and its empirical performance.

In this thesis we demonstrated the applicability of the two aforementioned messages on the example of co-clustering. First, we described two high level tasks that can be solved with the help of co-clustering. The first is discriminative prediction, in which a label is predicted based on two categor-

ical parameters. Clustering of the values of the parameters as an intermediate step improves the predictions by amplifying the statistical reliability of the relation. The second example is density estimation, in which the joint probability distribution of two categorical parameters is estimated. As previously, clustering the parameter values as an intermediate step can produce more reliable estimations. In the next step we analyzed the expected generalization performance of discriminative prediction and density estimation solutions based on co-clustering. Using the PAC-Bayesian bounds we showed by derivation that generalization performance depends on a tradeoff between empirical performance and the complexity of co-clustering, where the complexity of co-clustering is measured by the amount of mutual information that the hidden (cluster) variables preserve on the observed parameters. Finally, we suggested algorithms for optimization of this tradeoff. In the application of the suggested algorithms to the MovieLens dataset we demonstrated that the tradeoff has complete control over model regularization and that optimization of the tradeoff yields state-of-the-art results on this dataset.

A number of technical and conceptual novelties presented in this work worth are worthy of mention, in addition to the two main messages summarized above:

- In Section 3.3 we extended the PAC-Bayesian theorem for density estimation of a discrete function of the observations.
- Our approach to formulating the co-clustering objective underscores the notion of generalization in this task. In turn, this enables analysis of generalization properties of co-clustering solutions and the regularization of co-clustering models.
- In Section 4.3 we introduced the use of combinatorial priors in PAC-Bayesian bounds. These priors yield mutual information regularization terms, as opposed to L_2 and L_1 norm regularization resulting from Gaussian and Laplacian priors, respectively.
- We showed that the PAC-Bayesian approach is able to utilize the factor

form of prediction models and can be extended to graphical models beyond co-clustering.

8.1.1 The Meaning of Structures with Good Generalization Properties

As already stated, when learning a structure it is important to identify the high level goal for which it its intended. The main example of a high level goal considered in this thesis was the prediction of labels or events. It should be noted that structures that are good by this criterion are different from structures that are good by, for example, the stability criterion. This is best shown by the following example. Assume points in $\mathbb{R}^2$ are generated according to the following process. First, we select a center μ of a Gaussian according to a uniform distribution on a unit circle in $\mathbb{R}^2$. Then we generate a point $x \sim \mathcal{N}(\mu, \sigma^2 I)$ according to a Gaussian distribution centered at μ with a covariance matrix $\sigma^2 I$ (where I is a 2 by 2 identity matrix) for a fixed σ. Given a sample generated according to the above process we can apply a mixture of Gaussians clustering in order to learn the generating distribution. Note that:

- Due to the symmetry in the generating process, the solution will always be unstable (the centers of Gaussians in the mixture of Gaussians model can move arbitrarily along the unit circle).

- By increasing the sample size and the number of Gaussians in the mixture of Gaussians model we can approximate the true data generating process arbitrarily well.

Hence, models with good generalization properties are not necessarily stable. This point should be kept in mind when using generalization as an evaluation criterion in structure learning.

8.2 Future Work

This thesis forms a solid basis for several directions for future research which are discussed next.

8.2.1 A new Form of Matrix Factorization

For $d = 2$ the prediction model based on co-clustering

$$q(Y|X_1, X_2) = \sum_{C_1,C_2} q(Y|C_1, C_2)q(C_1|X_2)q(C_2|X_2)$$

can be considered as a form of matrix factorization. More specifically, let D be an $n_1 \times n_2$ matrix, possibly sparse, with the values of Y observed for corresponding combinations of $\langle X_1, X_2\rangle$. (Recall that $n_i = |\mathcal{X}_i|$.) For example, D can be a viewers by movies collaborative filtering matrix holding the ratings. Then we can write:

$$D \approx L^T M R, \tag{8.1}$$

where

$$L = q(C_1|X_1)$$

is an $m_1 \times n_1$ matrix mapping (stochastically) X_1-s to their clusters C_1-s,

$$R = q(C_2|X_2)$$

is an $m_2 \times n_2$ matrix mapping (stochastically) X_2-s to their clusters C_2-s, and

$$M = Y(C_1, C_2)$$

is an $m_1 \times m_2$ matrix describing what happens in the cluster product space. In this model each partition cell $\langle C_1, C_2\rangle$ is assigned a single label $Y(C_1, C_2)$.

In this form of matrix factorization L and R are stochastic matrices and M is arbitrary. Algorithm 6.1 can be applied to find a locally optimal factorization. There are several advantages to such factorization over other

matrix factorization forms including singular value decomposition (SVD) [97, 39] and non-negative matrix factorization [60, 61]:

- It has a clear probabilistic interpretation.
- It naturally handles missing values.
- Overfitting can be controlled via the regularization parameter β.
- The generalization bounds derived for co-clustering apply to this form of matrix factorization. (Strictly speaking, the analysis provided here applies only if the number of possible values in M is fixed ahead of time, but it can be extended and this point can be relaxed.)

As we have already shown, this form of matrix factorization achieves state-of-the-art prediction performance on collaborative filtering of the MovieLens dataset. One interesting direction for future research is to compare the performance of this form of matrix factorization with other forms of matrix factorization on other practical tasks.

Another promising direction for future research is to apply this form of matrix factorization in tasks, where multiple related datasets are considered. For example, let D_1 be a collaborative filtering matrix, D_2 be a matrix of viewers by viewer properties and D_2 be a matrix of movies by movie properties. We can look for simultaneous factorizations such that:

$$D_1 \approx L_1^T M_1 R_1$$
$$D_2 \approx L_1^T M_2 R_2$$
$$D_3 \approx L_3^T M_3 R_1.$$

In other words, the clustering of viewers into viewer clusters is shared between factorization of D_1 and D_2 and clustering of movies into movie clusters is shared between factorization of D_1 and D_3.

The above case is frequent in bioinformatics, when multiple experiments with partial relations are considered. For example, Alter et. al. [3] applied generalized SVD (GSVD) to compare yeast and human cell-cycle gene

expression datasets. In their experiment it is natural to create separate systems of clusters for yeast and human genes, but a common system of clusters for the cell-cycle time points. As already pointed out, the suggested method of matrix factorization has several advantages over SVD (and, consequently, over GSVD). Hence, it would be interesting to apply this in practice.

8.2.2 Evaluation of Unsupervised Learning Methods based on their Generalization Properties

In this thesis we have shown that it is possible to identify a high level task that introduces the notion of generalization to the traditionally unsupervised task of co-clustering. This in turn makes it possible to conduct generalization analyses and suggest regularization and model order selection approaches. This approach to the formulation of unsupervised learning can be extended to other unsupervised tasks, including classical ones, such as the learning of mixtures of Gaussians and the learning of graphical models. For example, in the task of learning a mixture of Gaussians model we can assume that the data points are generated by some unknown probability distribution. Then the Gaussian mixture model should be evaluated by its ability to predict the positions of new points generated by the same probability distribution. Similarly, in the task of learning graphical models we can assume that the samples were generated by some unknown probability distribution. The goal of the graphical model is then to be able to predict new samples generated by the same distribution.

To make the analysis of Gaussaian mixture models possible and to extend the analysis of generalization in graphical models beyond the simple examples already mentioned in this thesis it is essential to generalize the results of our work in two directions described in the following two sections.

8.2.3 PAC-Bayesian Analysis of Continuous Loss Functions

It is important to extend the PAC-Bayesian analysis to continuous loss functions. This requires substitution of theorem 3.7 by some analog which will hold for functions with infinite domains. Clearly, some other restrictions

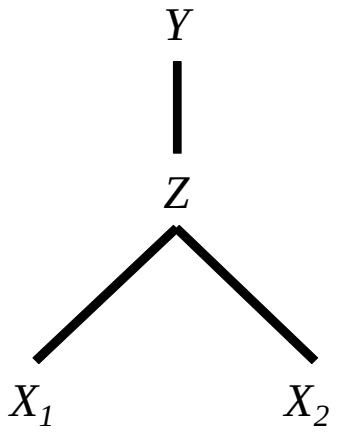


Figure 8.1: **Illustration of a graphical model corresponding to factorization in equation** (8.2)**.**

on the family of functions should be imposed. The extension to continuous loss functions has multiple applications, including regression and continuous density estimation. For example, it can be applied to analyze generalization in density estimation with Gaussian mixture models, as described earlier.

8.2.4 PAC-Bayesian Analysis of Generalization in Graphical Models

As described in Chapter 5, it is possible to generalize the PAC-Bayesian bounds for co-clustering to more complex graphical models. This suggests a new point of view on learning of graphical models: instead of learning graphical models that describe the data at hand we suggest learning graphical models with good generalization properties. A number of issues however would need to be tackled. They include:

- Development of efficient algorithms for learning the structure of a graphical model.

- Development of better bounds for some special forms of graphical models. For example, for the graphical model in Figure 8.1, which corre-

sponds to the following factorization model:

$$q(Y|X_1, X_2) = \sum_Z q(Y|Z)q(Z|X_1, X_2) \tag{8.2}$$
$$= \frac{q(X_1)q(X_2)}{q(X_1, X_2)} \sum_Z q(Y|Z)q(Z|X_1)q(Z|X_2)\frac{1}{q(Z)},$$

is it possible to derive a generalization bound that depends on the sum of mutual information $n_1\tilde{I}(Z; X_1) + n_2\tilde{I}(Z; X_2)$, but not on the size of the parameter product space $n_1 n_2 = |\mathcal{X}_1||\mathcal{X}_2|$?

8.2.5 When Structure Learning is Provably Superior?

I would like to conclude this work and the future work section with the question that was raised at the beginning and that still remained largely unresolved: "When structure learning is provably superior to other learning approaches?". In section 5.1 we proved that in classification by a single parameter learning the structure of that parameter does not help. We further proved that in classification by two or more parameters, clustering the values of the parameters improves the generalization properties of the predictions. It was not proved, however, that it is impossible to achieve similar or even better results using unstructured methods or methods based on implicit structure, like kernels.

The overarching question guiding the examination of structure learning in this work was prediction of some high level property. But this is by far from being the only high level task where structure learning can be useful. Examples of other high level tasks include computational efficiency, storage efficiency, robustness and control. The fact that we, as humans, perceive the world around us in a structured manner suggests that this mode of perception has some advantages over other possible ways we could have developed over the process of evolution. Identification and better understanding of these strong points of structure learning are important both for a better understanding of ourselves and for the formulation of better structure learning algorithms and comprehension of their outcomes.

Bibliography

[1] http://www.netflixprize.com/rules.

[2] http://www.grouplens.org.

[3] Orly Alter, Patrick O. Brown, and David Botstein. Generalized singular value decomposition for comparative analysis of genome-scale expression data sets of two different organisms. In *Proceedings of the National Academy of Science (PNAS)*, 2003.

[4] Amiran Ambroladze, Emilio Parrado-Hernández, and John Shawe-Taylor. Tighter PAC-Bayes bounds. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[5] S. Arimoto. An algorithm for computing the capacity of discrete memoryless channel. *IEEE Transactions on Information Theory*, 18, 1972.

[6] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[7] Jean-Yves Audibert and Olivier Bousquet. Combining PAC-Bayesian and generic chaining bounds. *Journal of Machine Learning Research*, 2007.

[8] Arindam Banerjee. On Bayesian bounds. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2006.

[9] Arindam Banerjee, Inderjit Dhillon, Joydeep Ghosh, Srujana Merugu, and Dhamendra Modha. A generalized maximum entropy approach to

Bregman co-clustering and matrix approximation. *Journal of Machine Learning Research*, 8, 2007.

[10] Peter Bartlett, Stéphane Boucheron, and Gábor Lugosi. Model selection and error estimation. *Machine Learning*, 2001.

[11] Peter Bartlett, Michael Collins, Ben Taskar, and David McAllester. Exponentiated gradient algorithms for large-margin structured classification. In *Advances in Neural Information Processing Systems (NIPS)*, 2005.

[12] Peter Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. In *Proceedings of the International Conference on Computational Learning Theory (COLT)*, 2001.

[13] Gill Bejerano, Yevgeny Seldin, Hanah Margalit, and Naftali Tishby. Markovian domain fingerprinting: statistical segmentation of protein sequences. *Bioinformatics*, 17(10):927–934, 2001.

[14] Shai Ben-David and Ulrike von Luxburg. Relating clustering stability to properties of cluster boundaries. In *Proceedings of the International Conference on Computational Learning Theory (COLT)*, 2008.

[15] Shai Ben-David, Ulrike von Luxburg, and David Pál. A sober look on clustering stability. In *Advances in Neural Information Processing Systems (NIPS)*, 2006.

[16] Yoshua Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2009. To appear.

[17] R. Blahut. Computation of channel capacity and rate distortion functions. *IEEE Transactions Information Theory*, 18:460–473, 1972.

[18] Gilles Blanchard and François Fleuret. Occam's hammer. In *Proceedings of the International Conference on Computational Learning Theory (COLT)*, 2007.

[19] Stéphane Boucheron, Gábor Lugosi, and Olivier Bousquet. Concentration inequalities. In O. Bousquet, U.v. Luxburg, and G. Rätsch, editors, *Advanced Lectures in Machine Learning*. Springer, 2004.

[20] Stéphane Boucheron, Gábor Lugosi, and Olivier Bousquet. Theory of classification: a survey of recent advances. *ESAIM: Probability and Statistics*, 2005.

[21] Olivier Catoni. PAC-Bayesian supervised classification: The thermodynamics of statistical learning. *IMS Lecture Notes Monograph Series*, 56, 2007.

[22] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[23] Yizong Cheng and George M. Church. Biclustering of expression data. In *Proceedings of the 8^{th} International Conference on Intelligent Systems for Molecular Biology (ISMB)*, 2000.

[24] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23, 1952.

[25] H. Cho and I. S. Dhillon. Co-clustering of human cancer microarrays using minimum sum-squared residue co-clustering. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 5:3, 2008.

[26] Hyuk Cho, Inderjit S. Dhillon, Yuqiang Guan, and Suvrit Sra. Minimum sum-squared residue co-clustering of gene expression data. In *Proceedings of the Fourth SIAM International Conference on Data Mining*, 2004.

[27] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications. John Wiley & Sons, New York, NY, 1991.

[28] Robert G. Cowell, A.Philip Dawid, Steffen L. Lauritzen, and David J. Spiegelhalter. *Probabilistic Networks and Expert Systems. Exact Computational Methods for Bayesian Networks.* Springer, 2007.

[29] Nello Cristianini and John Shawe Taylor. *An Introduction to Support Vector Machines.* Cambridge University Press, 2000.

[30] Philip Derbeko, Ran El-Yaniv, and Ron Meir. Explicit learning curves for transduction and application to clustering and compression algorithms. *Journal of Artificial Intelligence Research*, 22, 2004.

[31] Luc Devroye, László Györfi, and Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition.* Springer, 1996.

[32] Inderjot Dhillon, Subramanyam Mallela, and Dharmendra Modha. Information-theoretic co-clustering. In *ACM SIGKDD*, 2003.

[33] R. Duda, P. Hart, and D. Stork. *Pattern Classification.* John Wiley & Sons, New York, 2001.

[34] Miroslav Dudík, Steven J. Phillips, and Robert E. Schapire. Maximum entropy density estimation with generalized regularization and an application to species distribution modeling. *Journal of Machine Learning Research*, 8, 2007.

[35] Ran El-Yaniv and Oren Souroujon. Iterative double clustering for unsupervised and semi-supervised learning. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems (NIPS) 14.* MIT Press, 2001.

[36] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008.*, 2008.

[37] Dayane Freitag. Trained named entity recognition using distributional clusters. In *Proceedings of EMNLP*, 2004.

[38] Thomas George and Srujana Merugu. A scalable collaborative filtering framework based on co-clustering. In *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM05)*, 2005.

[39] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 3^{rd} edition, 1996.

[40] Peter Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007.

[41] J. A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337), 1972.

[42] Geremy Heitz, Stephen Gould, Ashutosh Saxena, and Daphne Koller. Cascaded classification models: Combining models for holistic scene understanding. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, 2009.

[43] Jonathan Herlocker, Joseph Konstan, Loren Terveen, and John Riedl. Evaluating collaborative filtering recommender systems. In *ACM Transactions on Information Systems*, volume 22(1), 2004.

[44] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[45] Thomas Hofmann. Probabilistic latent semantic analysis. In *UAI-1999*, 1999.

[46] Thomas Hofmann. Probabilistic latent semantic indexing. In *SIGIR-1999*, 1999.

[47] Anil Kumar Jain, M. Narasimha Murty, and Patrick Joseph Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3), 1999.

[48] Edwin Thompson Jaynes. Information theory and statistical mechanics. *Physical Review*, 106, 1957.

[49] Michael Kearns, Yishay Mansour, Andrew Ng, and Dana Ron. An experimental and theoretical comparison of model selection methods. *Machine Learning*, 1997.

[50] Michael J. Kearns and Umesh V. Vazirani. *An Introduction to Computational Learning Theory*. The MIT Press, 1994.

[51] Yuval Kluger, Ronen Basri, Joseph T. Chang, and Mark Gerstein. Spectral biclustering of microarray data: Coclustering genes and conditions. *Genome Research*, 2003.

[52] Fernando Pereira Koby Crammer, Mehryar Mohri. Gaussian margin machines. In *Proceedings on the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009.

[53] Vladimir Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, 2001.

[54] Eyal Krupka. *Generalization from Observed to Unobserved Features*. PhD thesis, The Hebrew University of Jerusalem, 2008.

[55] Eyal Krupka and Naftali Tishby. Generalization in clustering with unobserved features. In *Advances in Neural Information Processing Systems (NIPS)*, 2005.

[56] Eyal Krupka and Naftali Tishby. Generalization from observed to unobserved features by clustering. *Journal of Machine Learning Research*, 9, 2008.

[57] S. Kullback and R. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22, 1951.

[58] Tilman Lange, Volker Roth, Mikio L. Braun, and Joachim M. Buhmann. Stability based validation of clustering solutions. *Neural Computation*, 2004.

[59] John Langford. Tutorial on practical prediction theory for classification. *Journal of Machine Learning Research*, 2005.

[60] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401, 1999.

[61] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2001.

[62] Hang Li and Naoki Abe. Word clustering and disambiguation based on co-occurrence data. In *Proceedings of the 17th international conference on Computational linguistics*, 1998.

[63] David MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[64] Sara C. Madeira and Arlindo L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1):24–45, January 2004.

[65] Yishay Mansour and David McAllester. Generalization bounds for decision trees. In *Proceedings of the International Conference on Computational Learning Theory (COLT)*, 2000.

[66] Andreas Maurer. A note on the PAC-Bayesian theorem. www.arxiv.org, 2004.

[67] David McAllester. Some PAC-Bayesian theorems. *Machine Learning*, 37, 1999.

[68] David McAllester. PAC-Bayesian stochastic model selection. *Machine Learning*, 51(1), April 2003.

[69] David McAllester. Simplified PAC-Bayesian margin bounds. In *Proceedings of the International Conference on Computational Learning Theory (COLT)*, 2003.

[70] Tom Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[71] Liam Paninski. Estimation of entropy and mutual information. *Neural Computation*, 2003.

[72] François Laviolette Pascal Germain, Alexandre Lacasse and Mario Marchand. PAC-Bayesian learning of linear classifiers. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2009.

[73] Richard Rohwer and Dayne Freitag. Towards full automation of lexicon construction. In Dan Moldovan and Roxana Girju, editors, *HLT-NAACL 2004: Workshop on Computational Lexical Semantics*, 2004.

[74] Sivan Sabato and Shai Shalev-Shwartz. Prediction by categorical features: Generalization properties and application to feature ranking. In *Proceedings of the International Conference on Computational Learning Theory (COLT)*, 2007.

[75] Sivan Sabato and Shai Shalev-Shwartz. Ranking categorical features using generalization properties. *Journal of Machine Learning Research*, 9, 2008.

[76] Matthias Seeger. PAC-Bayesian generalization error bounds for Gaussian process classification. *Journal of Machine Learning Research*, 2002.

[77] Yevgeny Seldin. On unsupervised learning of mixtures of Markovian sources. Master's thesis, The Hebrew University of Jerusalem, 2001.

[78] Yevgeny Seldin, Gill Bejerano, and Naftali Tishby. Unsupervised segmentation and classification of mixtures of Markovian sources. In *Proceedings of the* 33^{rd} *Symposium on the Interface of Computing Science and Statistics*, 2001.

[79] Yevgeny Seldin, Gill Bejerano, and Naftali Tishby. Unsupervised sequence segmentation by a mixture of switching variable memory Markov sources. In *Proceedings of the* 18^{th} *International Conference on Machine Learning (ICML 2001)*, 2001.

[80] Yevgeny Seldin, Noam Slonim, and Naftali Tishby. Information bottleneck for non co-occurrence data. In B. Schölkopf, J. Platt, and

T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*. MIT Press, 2007.

[81] Yevgeny Seldin, Sonia Starik, and Michael Werman. Unsupervised clustering of images using their joint segmentation. In *Proceedings of the 3^{rd} International Workshop on Statistical and Computational Theories of Vision (SCTV 2003)*, 2003.

[82] Yevgeny Seldin and Naftali Tishby. Multi-classification by categorical features via clustering. In Andrew McCallum and Sam Roweis, editors, *Proceedings of the 25th Annual International Conference on Machine Learning (ICML 2008)*, 2008.

[83] Yevgeny Seldin and Naftali Tishby. PAC-Bayesian generalization bound for density estimation with application to co-clustering. In *Proceedings on the 12th International Conference on Artificial Intelligence and Statistics (AIStats 2009)*, 2009.

[84] Ohad Shamir, Sivan Sabato, and Naftali Tishby. Learning and generalization with the information bottleneck. In *Proceeding of the International Symposium on AI and Mathematics (ISAIM-2008)*, 2008.

[85] Ohad Shamir and Naftali Tishby. Cluster stability for finite samples. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[86] Ohad Shamir and Naftali Tishby. Model selection and stability in k-means clustering. In *Proceedings of the International Conference on Computational Learning Theory (COLT)*, 2008.

[87] Ohad Shamir and Naftali Tishby. On the reliability of clustering stability in the large sample regime. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[88] John Shawe-Taylor and Alex Dolia. A framework for probability density estimation. In *Proceedings of the 11^{th} International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2007.

[89] John Shawe-Taylor and David Hardoon. Pac-bayes analysis of maximum entropy classification. In *Proceedings on the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009.

[90] Noam Slonim. *The Information Bottleneck: Theory and Applications*. PhD thesis, The Hebrew University of Jerusalem, 2002.

[91] Noam Slonim, Shai Fine, and Naftali Tishby. Desciminative variable memory markov model for feature selection. In *Submitted to ICML 2001*, January 2001.

[92] Noam Slonim, Nir Friedman, and Naftali Tishby. Multivariate information bottleneck. *Neural Computation*, 18, 2006.

[93] Noam Slonim and Naftali Tishby. Document clustering using word clusters via the information bottleneck method. In *The 23^{rd} Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2000.

[94] Nathan Srebro. *Learning with Matrix Factorizations*. PhD thesis, MIT, 2004.

[95] Nathan Srebro, Noga Alon, and Tommi S. Jaakkola. Generalization error bounds for collaborative prediction with low-rank matrices. In *Advances in Neural Information Processing Systems (NIPS) 17*, 2005.

[96] Nathan Srebro, Jason Rennie, and Tommi Jaakkola. Maximum margin matrix factorization. In *Advances in Neural Information Processing Systems (NIPS) 17*. MIT Press, 2005.

[97] Gilbert Strang. *Introduction to linear algebra*. Wellesley-Cambridge Press, 4^{th} edition, 2009.

[98] Hiroya Takamura and Yuji Matsumoto. Co-clustering for text categorization. *Information Processing Society of Japan Journal*, 2003.

[99] Naftali Tishby, Fernando Pereira, and William Bialek. The information bottleneck method. In *Allerton Conference on Communication, Control and Computation*. 1999.

[100] L. G. Valiant. A theory of the learnable. *Communications of the Association for Computing Machinery*, 27(11), 1984.

[101] V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Soviet Math. Dokl.*, 9, 1968.

[102] V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2), 1971.

[103] V. N. Vapnik and A. Ya. Chervonenkis. Theory of pattern recognition. *Nauka, Moscow (in Russian)*, 1974. German translation: W.N.Wapnik, A.Ya.Tschervonenkis (1979), Theorie der Zeichenerkennug, Akademia, Berlin.

[104] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley Series on Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley & Sons, New York, NY, 1998.

[105] Ulrike von Luxburg and Shai Ben-David. Towards a statistical theory of clustering. In *PASCAL Workshop on Statistics and Optimization of Clustering*, 2005.

[106] Riu Xu and Donald Wunsch II. Survey of clustering algorithms. *IEEE transactions on neural networks*, 16(3), 2005.