A PAC-Bayesian Approach to Unsupervised Learning with Application to Co-clustering Analysis

Yevgeny Seldin
Max Planck Institute for Biological Cybernetics
Tübingen, Germany
and
School of Computer Science and Engineering
The Hebrew University of Jerusalem, Israel

Naftali Tishby
School of Computer Science and Engineering
Interdisciplinary Center for Neural Computation
The Hebrew University of Jerusalem, Israel

Abstract

This paper promotes a novel point of view on unsupervised learning. We argue that the goal of unsupervised learning is to facilitate a solution of some higher level task and it should be evaluated based on its contribution to the solution of that task. We present an example of such analysis in the case of co-clustering, which is a widely used approach in the analysis of data matrices. This paper identifies two possible high-level tasks in matrix data analysis: discriminative prediction of the missing entries and estimation of the joint probability distribution of row and column variables. For these two tasks we derive PAC-Bayesian generalization bounds for the expected out-of-sample performance of co-clustering-based solutions. The analysis yields regularization terms that were absent in the preceding formulations of co-clustering. The bounds suggest that the expected performance of co-clustering is governed by a trade-off between its empirical performance and the mutual information preserved by the cluster variables on row and column IDs. We derive an iterative projection algorithm for finding a local optimum of this trade-off. The algorithm achieved state-of-the-art performance in the MovieLens collaborative filtering task. The paper also features a number of important technical contributions:

- We derive a PAC-Bayesian bound for discrete density estimation.
- We introduce combinatorial priors to PAC-Bayesian analysis. They are appropriate for discrete optimization domains and lead to regularization terms in the form of mutual information.
- Co-clustering can be viewed as a Stochastic-Form Matrix Factorization (SFMF) $A \approx LMR$, where $L$ and $R$ are stochastic matrices and $M$ is arbitrary. SFM has a clear probabilistic interpretation. The generalization bound and the algorithm for finding a locally optimal solution derived for co-clustering are applicable to SFM.
- It is shown that PAC-Bayesian analysis of co-clustering can be extended to tree-shaped directed and undirected graphical models.

1. This paper is based on (Seldin and Tishby, 2008, 2009; Seldin, 2009).

©200? Yevgeny Seldin and Naftali Tishby.
1. Introduction

In many real-life situations the amount of supervision available for analysis of given data is limited or even non-existent. Even when present, supervision is often given at a high level, whereas the data are represented at a low level. For example, we can infer from an image label that it is e.g., an image of a boy with a cow, but the algorithm still has to locate the boy and the cow within a raw matrix of pixels. Nevertheless, many studies have shown that even completely unsupervised learning methods are able to identify meaningful structures in the data and can facilitate high-level decisions. But despite their remarkable success in practice, the 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 overlap 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, and cannot be used directly to improve existing solutions or to design new ones. The lack of solid theoretical foundations has given rise to multiple heuristic approaches that provide no guarantees on their expected performance on new data and therefore are hard to compare.

In this paper we reconsider the basic reasons for unsupervised learning and suggest new well-founded approaches to the evaluation and design of unsupervised structure learning algorithms. We argue that one does not learn structure just for the sake of it, but rather to facilitate the solution of some higher level task. By evaluation of the contribution of structure learning to the solution of the higher level task it is possible to derive an objective comparison of the utility of different structures in the context of that specific task. To be concrete, consider the following example: assume we have a set of cubes which we can cluster according to multiple parameters, such as shape, color, material they are made of, and so on. (Here we consider clustering as a simple example of a structure in the space of cubes.) All these structures (clusterings) co-exist simultaneously. Asking whether the clustering of cubes by shape is better or worse than the clustering of cubes by color seems at first to be analogous to comparing apples with oranges. However, if we say that after clustering the cubes we will have to pack them into a toy-car, then the clustering of cubes by shape is much more useful than the clustering of cubes by color, since packing is indifferent to color. We can further measure the amount of time that different clusterings saved us in the packing task and thereby produce in this context an objective numeric evaluation of the utility of clustering of cubes by different parameters.

Since in any set of non-trivial data many structures simultaneously co-exist, “blind” unsupervised learning without specification of the expected context of its potential application is doomed to failure in the general case. This is because the potential application (or range of applications) can make any property or element of the structure decisive or rather completely irrelevant for performing the task, and hence render it useful or useless for being identified by unsupervised learning. The necessity to consider unsupervised learning,
particularly, clustering within the context of its subsequent application was pointed out by many researchers, especially those concerned with practical applications of these methods (Guyon et al., 2009). In this paper we analyze the contribution of structure learning to a solution of a higher level task in the problem of 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 (Banerjee et al., 2007). Here we focus solely on co-clustering solutions that result in a grid form partition of the data matrix, as shown in Figure 1. This form of co-clustering is also known as partitional co-clustering (Banerjee et al., 2007), checkerboard bi-clustering (Cheng and Church, 2000; Kluger et al., 2003), grid clustering (Devroye et al., 1996; Seldin and Tishby, 2008, 2009), 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 (Hartigan, 1972; Madeira and Oliveira, 2004), but these forms of partitions are not discussed in this work. Note as well that this paper considers soft assignments of rows and columns to their clusters, as opposed to common co-clustering approaches which are restricted to hard assignments (Dhillon et al., 2003; Banerjee et al., 2007). 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 (Slonim and Tishby, 2000; El-Yaniv and Souroujon, 2001; Dhillon et al., 2003; Takamura and Matsumoto, 2003), genes and experimental conditions in bioinformatics (Cheng and Church, 2000; Cho et al., 2004; Kluger et al., 2003; Cho and Dhillon, 2008), tokens and contexts in natural language processing (Freitag, 2004; Rohwer and Freitag, 2004; Li and Abe, 1998), viewers and movies in recommender systems (George and Merugu, 2005; Seldin et al., 2007; Seldin, 2009), etc. In (Seldin et al., 2007; Seldin and Tishby, 2009) it was pointed out that there are actually two different classes of problems that are solved with co-clustering that correspond to two different high-level tasks and should be analyzed separately. The first class of problems are discriminative prediction tasks from which typical representative is collaborative filtering (Herlocker et al., 2004). In collaborative filtering 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. We assume that there is some unknown and unrestricted probability distribution \( p(X_1, X_2, Y) \) over the triplets of viewer \( X_1 \), movie \( X_2 \), and rating \( Y \). The goal is to build a discriminative predictor \( q(Y | X_1, X_2) \) that given a pair of viewer \( X_1 \) and movie \( X_2 \) will predict the expected rating \( Y \). A natural form of evaluation of such predictors, no matter whether they are based on co-clustering or not, is to evaluate the expected loss 
\[
\mathbb{E}_{p(X_1, X_2, Y)} \mathbb{E}_{q(Y' | X_1, X_2)} l(Y, Y'),
\]
where \( l(Y, Y') \) is an externally provided loss function for predicting \( Y' \) instead of \( Y \). In section 3 we provide this analysis for co-clustering-based predictors. The analysis enables not only to construct co-clustering solutions to this problem, but also to conduct a theoretical comparison of the co-clustering-based approach to this problem with other possible approaches.

The second class of problems, which are solved using co-clustering, are problems of estimation of a joint probability distribution in co-occurrence data analysis. A typical example of this kind of problems is the analysis of words-documents co-occurrence matrices in text mining (Slonim and Tishby, 2000; El-Yaniv and Souroujon, 2001; Dhillon et al., 2003). Words-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) have to be re-normalized.

Although many researchers have analyzed co-occurrence data by clustering similar words and similar documents (Slonim and Tishby, 2000; El-Yaniv and Souroujon, 2001; Dhillon et al., 2003; Takamura and Matsumoto, 2003), or by using probabilistic Latent Semantic Analysis (pLSA) and probabilistic Latent Semantic Indexing (pLSI) (Hofmann, 1999a,b) and other approaches, no clear learning task in this problem has been defined and it was hard to compare different approaches and to perform model order selection. In (Seldin and Tishby, 2009) one possible way of defining a high-level task for this problem was suggested. It was assumed that the observed co-occurrence matrix was drawn from an unknown and unrestricted joint probability distribution \( p(X_1, X_2) \) of words \( X_1 \) and documents \( X_2 \). The suggested task was estimation of this joint probability distribution based on the observed sample. In such formulation the quality of an estimator \( q(X_1, X_2) \) for \( p(X_1, X_2) \) can be measured by 
\[
-\mathbb{E}_{p(X_1, X_2)} \ln q(X_1, X_2),
\]
wherein 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 \( p(X_1, X_2) \) (Cover and Thomas, 1991). In section 3 we provide an analysis of this quantity for co-clustering-based density estimators. Similar to the case with co-clustering-based discriminative predictors, the analysis enables to perform model order selection in this problem. It further enables theoretical comparison of co-clustering-based approach to this problem with other possible approaches.
For the purpose of analysis and derivation of generalization bounds for the above two problems we found it convenient to apply the PAC-Bayesian framework (McAllester, 1999, 2003a). Similar to the Probably Approximately Correct (PAC) learning model (Valiant, 1984), PAC-Bayesian bounds pose no assumptions or restrictions on the distribution that generated the data (apart from the usual assumption that the data are independent identically distributed (i.i.d.) and that the train and test distributions are the same). However, unlike usual PAC bounds, wherein the whole hypothesis space is characterized by its VC-dimension (Vapnik, 1998), PAC-Bayesian bounds apply non-uniform treatment of the hypotheses by introducing a prior partition of the hypothesis space. For example, the class of decision trees can be divided into subclasses corresponding to tree depth and a preference to shallow trees can be given. If one manages to design a good partition of the hypothesis space, the tightness of the bounds can be improved considerably. In some cases the PAC-Bayesian bounds are only by 10%-20% far away from the test error, which makes them applicable in practice (Langford, 2005; Seldin and Tishby, 2008).

Originally the PAC-Bayesian bounds were derived for classification tasks. They were applied in the analysis of decision trees (Mansour and McAllester, 2000), Support Vector Machines (SVMs) (Langford and Shawe-taylor, 2002; McAllester, 2003b; Langford, 2005), transductive learning (Derbeko et al., 2004), and other domains. In section 2 we review the PAC-Bayesian bounds and present their extension to discrete density estimation tasks (first proposed in (Seldin and Tishby, 2009)). We further apply the PAC-Bayesian bounds to derive generalization bounds for discriminative prediction and density estimation with co-clustering in section 3. According to the derived bounds, the generalization performance of co-clustering-based models depends on a trade-off between their empirical performance and the mutual information that the clusters preserve on the observed parameters (row and column IDs). The mutual information term introduces model regularization that was missing in the previous formulations of co-clustering (Dhillon et al., 2003; Banerjee et al., 2007). We further suggest algorithms for optimization of the trade-off in section 5. In section 6 by optimization of the trade-off we achieve state-of-the-art performance in prediction of missing ratings in the MovieLens collaborative filtering dataset.

In section 4 we note that co-clustering can be regarded as a simple graphical model and suggest how to extend our analysis to more general graphical models. This provides a new perspective on learning graphical models. Instead of learning a graphical model that fits the training data, the approach suggests to optimize the model’s ability to predict new observations. We point out that graphical models can be naturally differentiated by their complexity and that PAC-Bayesian bounds are a handful tool that can utilize this heterogeneity to yield better bounds. In sections 3 and 4 it is demonstrated that PAC-Bayesian bounds are able to utilize the factor form of graphical models and provide bounds that depend on the sizes of the cliques of graphical models rather than the size of the whole parameter space.

In section 7 it is shown that co-clustering can be considered as a form of matrix factorization $A = L^TMR$, where $L$ and $R$ are stochastic matrices and $M$ is arbitrary. We term this form of factorization a Stochastic-Form Matrix Factorization (SFMF). The matrices $L$ and $R$ represent a stochastic (soft) assignment of rows and columns of a matrix to row and column clusters and $M$ provides an approximation of the values of $A$ in the cluster product space. This extends the preceding work of Banerjee et al. (2007), where
hard assignments were considered. One of the advantages of SFMF is its clear probabilistic interpretation. The generalization bounds and optimization algorithms developed for co-clustering are applicable to SFMF.

2. PAC-Bayesian Generalization Bounds

PAC-Bayesian generalization bounds are used as the main tool for the analysis of structure learning in this paper. This section reviews some known and presents some new PAC-Bayesian generalization bounds. For the audience that is less familiar with the PAC-Bayesian analysis, a simpler version of the bound called Occam’s razor, that holds for countable hypothesis spaces, can be found in (Seldin, 2009). It can be helpful to gain initial intuition about the PAC-Bayesian approach.

The PAC-Bayesian generalization bounds were suggested by McAllester (1999, 2003a) and build upon the classical PAC learning model (Valiant, 1984). The PAC learning model evaluates learning algorithms by their ability to predict new events coming from the same probability distribution that was used to train the algorithm. No restrictions on the data generating probability distribution are imposed except the assumption that the samples are i.i.d. Usually PAC bounds are derived by covering the error space of a hypothesis class. The most familiar PAC bounds were based on the Vapnik-Chervonenkis (VC) dimension of a hypothesis class (Vapnik and Chervonenkis, 1968, 1971; Vapnik, 1998; Devroye et al., 1996), whereas more recent bounds involve Rademacher and Gaussian complexities (Koltchinskii, 2001; Bartlett et al., 2001; Bartlett and Mendelson, 2001; Boucheron et al., 2005). However, in all the above approaches the whole hypothesis class is characterized by a single number: its VC-dimension or Rademacher complexity, which means that all the incorporating hypotheses are treated identically. PAC-Bayesian bounds are derived by covering the hypothesis space and they introduce non-uniform treatment of the hypotheses. In PAC-Bayesian approach each hypothesis is characterized by its own complexity defined by its prior. This refined approach enables to tighten the bounds considerably and make them meaningful in a practical sense: in some applications the discrepancy between the bound value and the test error is only 10%-20% (Langford, 2005; Seldin and Tishby, 2008; Seldin, 2009). There is one more distinction between the usual PAC analysis and the PAC-Bayesian bounds that extend the scope of applicability of the latter. Classical PAC analysis aims in bounding the discrepancy between the expected performance of the hypothesis with the best empirical performance, and the best expected performance that could be achieved within the given hypothesis class. We call these type of bounds regret bounds. To derive a regret bound one should bound the best expected performance within a hypothesis class, which can be done only if the hypothesis class has a finite VC-dimension. PAC-Bayesian bounds bound the expected performance of a given hypothesis, but do not attempt to bound the distance to the best it could be. This fact enables us to apply PAC-Bayesian bounds even in situations where the VC-dimension of a hypothesis class is infinite, for example, decision trees of unlimited depth or separating hyperplanes in infinite-dimensional spaces. In most practical applications it is sufficient to bound the expected performance of the obtained classifier and it is not essential to derive a regret bound. For example, if we are assured that a given diagnosis tool performs correctly in 99% of the cases, it may be sufficient for our needs even if we do not know what the best performance is that we could achieve. This
does not disprove the basic result of the PAC learning theory, which states that learning is possible if and only if the VC-dimension of a hypothesis class is finite, but rather extends the notion of learnability. Instead of regret-based definition of learnability, by which the ability to learn is the ability to achieve, up to a small epsilon, the best possible solution within a hypothesis class, PAC-Bayesian approach defines learnability as the ability to bound the expected performance of the obtained solution (and then it’s a question of whether this expected performance is sufficient for our needs).

Since the strength of PAC-Bayesian analysis lies in its ability to provide non-uniform treatment of the hypotheses in a hypothesis class its advantage over traditional PAC analysis is most prominent in analysis of heterogeneous hypothesis classes (or, in other words, when the hypotheses constituting a hypothesis class are not symmetric). Some hypothesis classes exhibit a “natural heterogeneity”, for example, we can partition the class of all possible decision trees of unlimited depth into subclasses according to the tree depth. In some other cases it can be possible to introduce 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 (McAllester, 2003b; Langford, 2005). In structure learning the hypothesis class usually exhibits a natural heterogeneity since the hypotheses (structures) can be differentiated by their complexity. Hence, PAC-Bayesian analysis has a great potential in the analysis of structure learning, which is only partially explored in this work. PAC-Bayesian bounds are further distinguished by their explicit dependence on model parameters, which makes them easy to apply in optimization.

Following the pioneering work of McAllester (1999) several minor improvements to the PAC-Bayesian bound (Audibert and Bousquet, 2007; Blanchard and Fleuret, 2007; Maurer, 2004; Germain et al., 2009) as well as some simplifications to its proof (Seeger, 2002; Langford, 2005; Maurer, 2004; Banerjee, 2006) were suggested. The proof presented here draws on the works of Maurer (2004) and Banerjee (2006) and is slightly tighter and simpler than the original proof in (McAllester, 2003a). One of the most extensively studied applications of the PAC-Bayesian bound is the analysis of SVMs (McAllester, 2003a; Langford, 2005; Ambroladze et al., 2007; Koby Crammer, 2009; Germain et al., 2009). When a Gaussian prior over the linear separators is selected, the bounds provide theoretical justification for the maximum margin principle in learning SVMs. They are also the tightest known bounds for SVMs. Derbeko et al. (2004) applied PAC-Bayesian bounds to analysis of transduction learning. Other applications include maximum margin analysis of structured prediction (Bartlett et al., 2005; McAllester, 2007). In (Seldin and Tishby, 2008) we suggested applying the PAC-Bayesian bound to the analysis of co-clustering which is presented in section 3.

The initial work of McAllester (1999) as well as all subsequent works on PAC-Bayesian bounds were concerned with classification scenario. In (Seldin and Tishby, 2009) we extended the PAC-Bayesian framework and applied it to discrete density estimation. In order to present the bounds for classification and density estimation we have to define the notion of randomized predictors, which is done next. After the definition we present the PAC-Bayesian theorems and their proofs.
2.1 Randomized Predictors

Let $\mathcal{H}$ be a hypothesis class and let $S$ be an i.i.d. sample of size $N$. For each $h \in \mathcal{H}$ we denote by $\hat{L}(h)$ the empirical loss of the hypothesis $h$ on $S$ and by $L(h)$ the expected loss of $h$ with respect to the true, unknown and unrestricted probability that generates the data.

Let $\mathcal{Q}(h)$ be a distribution over $\mathcal{H}$. 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 $x$. In classification context, $\mathcal{Q}$ is termed a randomized classifier (Langford, 2005). However, since this work extends the PAC-Bayesian framework beyond the classification scenario by using the same randomization technique we use the term “randomized predictor”. In this more general context $h(x)$ is a general function of $x$ and not necessarily a classifier.

We further extend the definitions of the empirical and expected losses for randomized predictors in the following way:

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

(1)

and

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

(2)

For two distributions $\mathcal{Q}$ and $\mathcal{P}$ over $\mathcal{H}$ we define

$$D(\mathcal{Q}\|\mathcal{P}) = \mathbb{E}_{\mathcal{Q}(h)} \ln \frac{\mathcal{Q}(h)}{\mathcal{P}(h)}$$

(3)

to be the Kullback-Leibler (KL) divergence between $\mathcal{Q}$ and $\mathcal{P}$ (Cover and Thomas, 1991). As well, we define

$$D_b(\hat{L}(\mathcal{Q}) \| L(\mathcal{Q})) = \hat{L}(\mathcal{Q}) \ln \frac{\hat{L}(\mathcal{Q})}{L(\mathcal{Q})} + (1 - \hat{L}(\mathcal{Q})) \ln \frac{1 - \hat{L}(\mathcal{Q})}{1 - L(\mathcal{Q})}$$

(4)

to be the KL-divergence between two Bernoulli distributions with biases $\hat{L}(\mathcal{Q})$ and $L(\mathcal{Q})$. Now we are ready to state the PAC-Bayesian theorems.

2.2 PAC-Bayesian Theorems

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

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

(5)

Theorem 2 (PAC-Bayesian bound for discrete density estimation) 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 $h \in \mathcal{H}$ is a function from $\mathcal{X}$ to a finite set $Z$ with cardinality $|Z|$. Let $p_h(Z) = P_{X \sim p(X)} \{ h(X) = Z \}$ be the distribution over $Z$ 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 $p_Q(Z) = \mathbb{E}_{\mathcal{Q}(h)} p_h(Z)$ 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 $\hat{p}_h(Z) = P_{X \sim \bar{p}(X)}\{h(X) = Z\}$ be the empirical distribution over $Z$ corresponding to $h$ and $S$ and $\bar{p}_Q(Z) = E_{Q(h)}\hat{p}_h(Z)$. Then with a probability greater than $1 - \delta$ for all possible $Q$ simultaneously:

$$D(\hat{p}_Q(Z)||p_Q(Z)) \leq \frac{D(Q||P) + (|Z| - 1)\ln(N + 1) - \ln\delta}{N}. \quad (6)$$

Remarks:

1. The PAC-Bayesian bound for classification (5) is a direct consequence of the PAC-Bayesian bound for density estimation (6). In order to observe this, let $Z$ be the error variable. Then each hypothesis $h \in H$ is a function from the sample space (in this case the samples are pairs $(X, Y)$) to the error variable $Z$ and $|Z| = 2$. Furthermore, $\hat{L}(Q) = P_{\bar{p}(X,Y)}\{Z = 1\}$ and $L(Q) = P_{p(X,Y)}\{Z = 1\}$, hence $D_b(\hat{L}(Q)||L(Q)) = D(\hat{p}_Q(Z)||p_Q(Z))$. Substituting this into (6) yields (5).

2. Maurer (2004) has shown that due to convexity of the KL-divergence (5) is valid for all loss functions bounded in the $[0,1]$ interval, and not only for the zero-one losses. He also proved that due to tighter concentration of empirical means of binary variables for $N \geq 8$ bound (5) can be further tightened:

$$D_b(\hat{L}(Q)||L(Q)) \leq \frac{D(Q||P) + \frac{1}{2}\ln(4N) - \ln\delta}{N}. \quad (7)$$

3. The proof of theorem 2 presented below reveals a close relation between the PAC-Bayesian theorems and the method of types in information theory (Cover and Thomas, 1991). Further relations between the PAC-Bayesian bounds, information theory and statistical mechanics are discussed in (Catoni, 2007).

4. The trade-off between $\hat{L}(Q)$ and $D(Q||P)$ in the PAC-Bayesian bounds has also a tight relation to the maximum entropy principle in learning and statistical mechanics (Jaynes, 1957; Dudík et al., 2007; Catoni, 2007; Shawe-Taylor and Hardoon, 2009). This point is further discussed in (Catoni, 2007; Shawe-Taylor and Hardoon, 2009).

The proof of theorem 2 presented below is based on three simple steps. First we bound the expectation of the exponent of the divergence, $E_S e^{ND(\hat{p}_h(Z)||p_h(Z))}$ for a single hypothesis $h$. Then we relate the divergence $D(\hat{p}_Q(Z)||p_Q(Z))$ for all $Q$ to a single (prior) reference measure $P$. In this step we obtain that $ND(\hat{p}_Q(Z)||p_Q(Z)) \leq D(Q||P) + \ln E_P(h) e^{ND(\hat{p}_h(Z)||p_h(Z))}$. Finally, we apply the result from the first step to bound $E_S [E_P(h) e^{ND(\hat{p}_h(Z)||p_h(Z))}]$ and obtain (6). The first two steps of the proof have value on their own and therefore are presented in dedicated subsections.

2.3 The Law of Large Numbers

In this section we 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 (Cover and Thomas, 1991).
Theorem 3 Let $S = \{X_1, \ldots, X_N\}$ 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:

$$\mathbb{E}_S e^{ND(\hat{p}(X)||p(X))} \leq (N + 1)^{|X| - 1}. \quad (8)$$

**Proof** Enumerate the possible values of $X$ by $1, \ldots, |X|$ 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 $H(\hat{p}) = -\sum \hat{p}_i \ln \hat{p}_i$ be the empirical entropy. Then:

$$\mathbb{E}_S e^{ND(\hat{p}(X)||p(X))} = \sum_{n_1, \ldots, n_{|X|}: \sum_i n_i = N} \left( \begin{array}{c} N \\ n_1, \ldots, n_{|X|} \end{array} \right) \cdot \prod_{i=1}^{|X|} p_i^{N\hat{p}_i} \cdot e^{ND(\hat{p}(X)||p(X))} \leq \sum_{n_1, \ldots, n_{|X|}: \sum_i n_i = N} e^{N\sum_i \hat{p}_i \ln p_i} \cdot e^{ND(\hat{p}(X)||p(X))} \quad (9)$$

$$= \sum_{n_1, \ldots, n_{|X|}: \sum_i n_i = N} 1 = \left( \begin{array}{c} N + |X| - 1 \\ |X| - 1 \end{array} \right) \leq (N + 1)^{|X| - 1}. \quad (10)$$

In (9) we use the $\left( \begin{array}{c} N \\ n_1, \ldots, n_{|X|} \end{array} \right) \leq e^{NH(\hat{p})}$ bound on the multinomial coefficient, which counts the number of sequences with a fixed cardinality profile (type) $n_1, \ldots, n_{|X|}$ (Cover and Thomas, 1991). In the second equality in (10) 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. \[\blacksquare\]

**Corollary of Theorem 3**

Note in passing that it is straightforward to recover theorem 12.2.1 in (Cover and Thomas, 1991) from theorem 3. We even suggest a small improvement over it:

**Theorem 4 (12.2.1 in Cover and Thomas, 1991)** Under the notations of theorem 3:

$$P \{ D(\hat{p}(X)||p(X)) \geq \varepsilon \} \leq e^{-N\varepsilon + (|X| - 1)\ln(N + 1)}, \quad (11)$$

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

$$D(\hat{p}(X)||p(X)) \leq \frac{(|X| - 1)\ln(N + 1) - \ln \delta}{N}. \quad (12)$$

**Proof** By Markov’s inequality and theorem 3:

$$P \{ D(\hat{p}(X)||p(X)) \geq \varepsilon \} = P \{ e^{ND(\hat{p}(X)||p(X))} \geq e^{N\varepsilon} \} \leq \frac{\mathbb{E}_S e^{ND(\hat{p}(X)||p(X))}}{e^{N\varepsilon}} \leq \frac{(N + 1)^{|X| - 1}}{e^{N\varepsilon}} = e^{-N\varepsilon + (|X| - 1)\ln(N + 1)}. \quad \blacksquare$$
A Tighter Bound for Binary Variables

Maurer (2004) proved that for $|X| = 2$ and $N \geq 8$ a tighter bound holds:

$$E_s e^{ND(\hat{p}(X)\|p(X))} \leq 2\sqrt{N}. \quad (13)$$

This result is at the basis for the slightly tighter version (7) of the PAC-Bayesian bound for classification (5). Maurer (2004) also proved that for $N \geq 2$

$$E_s e^{ND(\hat{p}(X)\|p(X))} \geq \sqrt{N}, \quad (14)$$

hence the PAC-Bayesian bound for classification cannot be further tightened using the presented proof technique.

2.4 Change of Measure Inequality

Simultaneous treatment of all possible distributions (measures) $Q$ over $H$ is done by relating them all to a single reference (prior) measure $P$. We call this relation a change of measure inequality. It appears in the proof of the PAC-Bayesian theorem in (McAllester, 2003b) and was formulated as a standalone result in (Banerjee, 2006). Banerjee (2006) 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 5 (Change of Measure Inequality)** For any measurable function $\phi(h)$ on $H$ and any distributions $P$ and $Q$ on $H$, we have:

$$E_Q(h) \phi(h) \leq D(Q\|P) + \ln E_P(h) e^{\phi(h)}. \quad (15)$$

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

$$E_Q(h) \phi(h) = E_Q(h) \ln \left( \frac{dQ(h)}{dP(h)} \cdot e^{\phi(h)} \cdot \frac{dP(h)}{dQ(h)} \right)$$

$$= D(Q\|P) + E_Q(h) \ln \left( e^{\phi(h)} \cdot \frac{dP(h)}{dQ(h)} \right)$$

$$\leq D(Q\|P) + \ln E_P(h) \left( e^{\phi(h)} \cdot \frac{dP(h)}{dQ(h)} \right)$$

$$= D(Q\|P) + \ln E_P(h) e^{\phi(h)}, \quad (16)$$

where (16) is by Jensen’s inequality.

2.5 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.
Proof of Theorem 2 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:

\[
ND(\hat{p}_Q(Z) || p_Q(Z)) = ND(E_{Q(h)}\hat{p}_h(Z) || E_{Q(h)}p_h(Z)) \\
\leq E_{Q(h)}ND(\hat{p}_h(Z) || p_h(Z)) \\
\leq D(Q||P) + \ln E_{P(h)}e^{ND(\hat{p}_h(Z) || p_h(Z))}, \tag{17}
\]

where (17) is by the convexity of the KL-divergence (Cover and Thomas, 1991) and (18) is by the change of measure inequality. To obtain (6) it is left to bound \( E_{P(h)}e^{ND(\hat{p}_h(Z) || p_h(Z))} \):

\[
E_S \left[E_{P(h)}e^{ND(\hat{p}_h(Z) || p_h(Z))}\right] = E_{P(h)} \left[E_S e^{ND(\hat{p}_h(Z) || p_h(Z))}\right] \leq (N + 1)^{|Z|^{-1}}, \tag{19}
\]

where the last inequality is justified by the fact that \( E_S e^{ND(\hat{p}_h(Z) || p_h(Z))} \leq (N + 1)^{|Z|^{-1}} \) for each \( h \) individually according to (8). By (19) and Markov’s inequality we conclude that with a probability of at least \( 1 - \delta \) over \( S \):

\[
E_{P(h)}e^{ND(\hat{p}_h(Z) || p_h(Z))} \leq \frac{(N + 1)^{|Z|^{-1}}}{\delta}. \tag{20}
\]

Substituting this into (18) and normalizing by \( N \) yields (6).

\[\square\]

2.6 Construction of a Density Estimator

Although we have bounded \( D(\hat{p}_Q(Z) || p_Q(Z)) \) in theorem 2, \( \hat{p}_Q(Z) \) still cannot be used as a density estimator for \( p_Q(Z) \), because it is not bounded from zero. In order to bound the logarithmic loss \( -E_{p_Q(Z)} \ln \hat{p}_Q(Z) \), which corresponds e.g. to the expected code length of encoder \( \hat{p}_Q \) when samples are generated by \( p_Q \) (Cover and Thomas, 1991), we have to smooth \( \hat{p}_Q \). We denote a smoothed version of \( \hat{p}_Q \) by \( \tilde{p}_Q \) and define it as:

\[
\hat{p}_h(Z) = \frac{\hat{p}_h(Z) + \gamma}{1 + \gamma |Z|}, \tag{21}
\]

\[
\tilde{p}_Q(Z) = E_{Q(h)}\hat{p}_h(Z) = \frac{\hat{p}_Q(Z) + \gamma}{1 + \gamma |Z|}. \tag{22}
\]

In the following theorem we show that if \( D(\hat{p}_Q(Z) || p_Q(Z)) \leq \varepsilon(Q) \) and \( \gamma = \frac{\sqrt{\varepsilon(Q)/2}}{|Z|} \), then

\[-E_{p_Q(Z)} \ln \tilde{p}_Q(Z) \text{ is roughly within } \pm \sqrt{\varepsilon(Q)/2} \ln |Z| \text{ range around } H(\hat{p}_Q(Z)). \]

The bound on \( D(\tilde{p}_Q(Z) || p_Q(Z)) \) is naturally obtained by theorem 2. Thus, the performance of the density estimator \( \hat{p}_Q \) is optimized by distribution \( Q \) that minimizes the trade-off between \( H(\hat{p}_Q(Z)) \) and \( \frac{1}{N} D(Q||P) \).

Note that for a uniform distribution \( u(Z) = \frac{1}{|Z|} \) the value of \( -E_{p(Z)} \ln u(Z) = \ln |Z| \).

Thus, the theorem is interesting when \( \sqrt{\varepsilon(Q)/2} \) is significantly smaller than 1. For technical reasons in the proofs of the following section, the upper bound in the next theorem is stated for \(-E_{p_Q(Z)} \ln \hat{p}_Q(Z)\) and for \(-E_{Q(h)}E_{p h(Z)} \ln \hat{p}_h(Z)\). We also denote \( \varepsilon = \varepsilon(Q) \) for brevity.
Theorem 6 Let $Z$ be a random variable distributed according to $p_Q(Z)$ and assume that $D(\hat{p}_Q(Z) \| p_Q(Z)) \leq \varepsilon$. Then $-E_{p_Q(Z)} \ln \hat{p}_Q(Z)$ is minimized by $\gamma = \frac{\sqrt{\varepsilon/2}}{|Z|}$. For this value of $\gamma$ the following inequalities hold:

\begin{align}
- & E_{p_W(Z)} E_{p_n(Z)} \ln \hat{p}_h(Z) \leq H(\hat{p}_Q(Z)) + \sqrt{\varepsilon/2} \ln |Z| + \phi(\varepsilon), \\
- & E_{p_Q(Z)} \ln \hat{p}_Q(Z) \leq H(\hat{p}_Q(Z)) + \sqrt{\varepsilon/2} \ln |Z| + \phi(\varepsilon), \\
- & E_{p_Q(Z)} \ln \hat{p}_Q(Z) \geq H(\hat{p}_Q(Z)) - \sqrt{\varepsilon/2} \ln |Z| - \psi(\varepsilon),
\end{align}

where:

\[ \psi(\varepsilon) = \sqrt{\frac{\varepsilon}{2}} \ln \frac{1 + \sqrt{\varepsilon}}{\sqrt{\varepsilon}} \quad \text{and} \quad \phi(\varepsilon) = \psi(\varepsilon) + \ln(1 + \sqrt{\frac{\varepsilon}{2}}). \]

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

The proof is provided in appendix A.

3. PAC-Bayesian Analysis of Co-clustering

In the introduction we defined two high-level goals which can be solved via co-clustering. The first is discriminative prediction of the matrix entries in the analysis of functional data, such as collaborative filtering. The second is the estimation of joint probability distribution in co-occurrence data analysis. We further defined the notion of generalization for each of the two problems. In this section we apply the PAC-Bayesian generalization bounds in order two derive generalization bounds for the two settings. We begin with the co-clustering approach to discriminative prediction, which is slightly easier in terms of presentation. Then we consider the discrete density estimation problem.

3.1 PAC-Bayesian Analysis of Discriminative Prediction with Grid Clustering

Let $X_1 \times \ldots \times X_d \times Y$ be a $(d + 1)$-dimensional product space. We assume that each $X_i$ is categorical, its cardinality fixed and known, which we denote by $|X_i| = n_i$. We also assume that $Y$ 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 $d = 2$, $X_1$ is the space of viewers, $n_1$ is the number of viewers, $X_2$ is the space of movies, $n_2$ is the number of movies, and $Y$ is the space of the ratings (e.g., on a five-star scale). The loss $l(Y, Y')$ 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 either the space of viewers or on the space of movies; thus both $X_1$ and $X_2$ are categorical.

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

\[ q(Y|X_1, \ldots, X_d) = \sum_{C_1, \ldots, C_d} q(Y|C_1, \ldots, C_d) \prod_{i=1}^{d} q(C_i|X_i). \]
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 $(C_1, ..., C_d)$ in the cluster product space. The prediction model (26) corresponds to the graphical model in Figure 2.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 $Q = \{q(C_i|X_i)\}_{i=1}^d, q(Y|C_1, ..., C_d)\}$. In the next subsection we show that (26) corresponds to a randomized prediction strategy. We further denote:

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

and

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

where $q(Y|X_1, ..., X_d)$ is defined by (26).

We define

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

where $x_i \in X_i$ are the possible values of $X_i$, $c_i$ are the possible values of $C_i$, and

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

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

With the above definitions we can state the following generalization bound for discriminative prediction with co-clustering.
Theorem 7 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 \( \mathcal{Q} = \{\{q(C_i|X_i)\}_{i=1}^d, q(Y|C_1, .., C_d)\} \):

\[
D_b(\hat{L}(\mathcal{Q})||L(\mathcal{Q})) \leq \frac{\sum_{i=1}^d (n_i\bar{I}(X_i; C_i) + m_i \ln n_i) + M \ln |Y| + \frac{1}{2} \ln(4N) - \ln \delta}{N},
\]

where

\[
M = \prod_{i=1}^d m_i
\]

is the number of partition cells.

Remarks:

- Observe that given a prediction strategy \( \mathcal{Q} = \{\{q(C_i|X_i)\}_{i=1}^d, q(Y|C_1, .., C_d)\} \) both \( \hat{L}(\mathcal{Q}) \) and \( \bar{I}(X_i; C_i) \) are computable exactly.

- Any bounded loss greater than 1 can be normalized to the \([0,1]\) interval and the bound can still be applied.

- For purposes of the discussion below it is easier to look at the weaker, but explicit form of the bound (31):

\[
L(\mathcal{Q}) \leq \hat{L}(\mathcal{Q}) + \sqrt{\frac{\sum_{i=1}^d (n_i\bar{I}(X_i; C_i) + m_i \ln n_i) + M \ln |Y| + \frac{1}{2} \ln(4N) - \ln \delta}{2N}}.
\]

The bound (33) follows from (31) by the \( L_1 \)-norm lower bound on the KL-divergence (Cover and Thomas, 1991) which states that for two Bernoulli variables with biases \( p \) and \( q \)

\[
D_b(p||q) \geq 2(p - q)^2.
\]

Discussion: There are two cases in the collaborative filtering task that provide a good intuition on the co-clustering approach to this problem. If we assign all of 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 given that 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 trade-off between locality of the predictions and their statistical reliability.

This trade-off is explicitly exhibited in the bound (33). If we assign all \( X_i \)-es to a single cluster, then \( \bar{I}(X_i; C_i) = 0 \) 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 \( \bar{I}(X_i; C_i) \) is large, specifically in this case \( \bar{I}(X_i; C_i) = \ln n_i \), and \( \hat{L}(\mathcal{Q}) \) is far from \( L(\mathcal{Q}) \). But there are even finer observations we can draw from the bound. Bear in mind that \( n_i\bar{I}(X_i; C_i) \) is linear in \( n_i \), whereas \( m_i \ln n_i \) is logarithmic in \( n_i \).
Figure 3: 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 \(\binom{2}{4} = 6\) 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).

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 (33) is \(n_i \hat{I}(X_i; C_i)\). This term penalizes for the effective complexity of a partition, rather than the raw number of clusters used. For example, the unbalanced partition of a 4×4 matrix into 2×2 clusters in Figure 3.a is simpler than the balanced partition into the same number of clusters in Figure 3.b. The reason, which will become clearer after we have defined the prior over the space of partitions in subsection 3.3, is that there are fewer unbalanced partitions than balanced ones. Therefore, the subspace of unbalanced partitions is smaller than the subspace of balanced partitions and it is easier to describe an unbalanced partition rather than a balanced one. Intuitively, the partition in Figure 3.a is not fully utilizing the 2×2 clusters that it could use, and should therefore be penalized less. At a practical level, the bound enables us at the optimization step to operate with more clusters than are actually required and to penalize the final solution according to the measure of cluster utilization. This claim is supported by our experiments. To summarize this point, the bound (33) suggests a trade-off between the empirical performance and the effective complexity of a partition.

Finally, consider the \(M \ln |Y|\) term in the bound. \(M = (\prod_{i=1}^{d} m_i)\) is the number of partition cells (in a hard partition) and \(M \ln |Y|\) corresponds to the size of the \(\langle C_1, ..., C_d, Y \rangle\) clique in Figure 2.a. The number of sample points \(N\) 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 section 4.

Proof of Theorem 7 The proof of theorem 7 is a direct application of the PAC-Bayesian bound for classification in theorem 1 (or more precisely, of its refinement in (7)). 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 subsection and design a prior over it in subsection 3.3. Then, substitution of the calculation of $D(\mathcal{Q}\|\mathcal{P})$ in lemma 9 into theorem 1 completes the proof.

3.2 Grid Clustering Hypothesis Space

The hypothesis space $\mathcal{H}$ with which we select to work is the space of hard grid partitions of the product space $\mathcal{X}_1 \times \ldots \times \mathcal{X}_d$ (as illustrated in Figure 3) augmented with label assignments to the partition cells. (In subsection 3.4 we use grid partitions without labels on the partition cells, thus the discussion in this and the following subsection is kept 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 $c_i \in \{1, \ldots, m_i\}$. To work with $\mathcal{H}$ we use the following notations:

- Let $\bar{m} = (m_1, \ldots, m_d)$ 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 $i$.
- Let $\mathcal{H}_{\bar{m}}$ denote the subspace of partitions of $\mathcal{X}_1 \times \ldots \times \mathcal{X}_d$ in which the number of clusters used along each dimension matches $\bar{m}$. Obviously, for distinct $\bar{m}$-s, $\mathcal{H}_{\bar{m}}$-s are disjoint.
- We use $\mathcal{H}|_{y|\bar{m}}$ to denote the space of possible assignments of labels to $\mathcal{H}_{\bar{m}}$. Then we can write $\mathcal{H} = \bigcup_{\bar{m}} (\mathcal{H}_{\bar{m}} \times \mathcal{H}|_{y|\bar{m}})$.
- For each $h \in \mathcal{H}$ we write $h = h|_{1} \times \ldots \times h|_{d} \times h|_{y|\bar{m}}$, 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 $h = h|_{1} \times \ldots \times h|_{d}$, without the labels assignment.

It should be pointed out that $\mathcal{Q} = \{q(C_i|X_i)\}_{i=1}^d, q(Y|C_1, \ldots, C_d)\}$ is a distribution over $\mathcal{H}$ and (26) 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 $\{q(C_i|X_i)\}_{i=1}^d, q(Y|C_1, \ldots, C_d)\}$. 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 $q(Y|C_1, \ldots, C_d)$. For example, we map each viewer to a cluster of viewers, 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 $\langle x_1, \ldots, x_d \rangle$ we just check into which partition cell it has fallen 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 (26) 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 \( \langle X_1, \ldots, X_d \rangle \) directly. Nevertheless, (26) corresponds to the randomized prediction process described above. This makes it possible to apply the PAC-Bayesian analysis.

### 3.3 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 \( D(Q\|P) \) 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 the calculations of the KL-divergence. This can be contrasted with the \( L_2 \)-norm and \( L_1 \)-norm terms resulting from Gaussian and Laplacian priors respectively in the analysis of SVMs (McAllester, 2003b). Another important point to mention is that the posterior \( Q \) returns a named partition of \( X_i \)'s. However, the hypothesis space \( H \) and the prior \( P \) 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. In this way we account for all possible name permutations that are irrelevant for the solution.

The statements in the next two lemmas are given in two versions, one for \( H \) augmented with labels, which is used in the proofs of theorem 7, and the other one for \( H_{\bar{m}} \) without the labels, which is used later for the proofs on density estimation with grid clustering.

**Lemma 8** It is possible to define a prior \( P \) over \( H_{\bar{m}} \) that satisfies:

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

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 \( P \) over \( H = \bigcup_{\bar{m}} (H_{\bar{m}} \times H|_{\bar{m}}) \) that satisfies:

\[
P(h) \geq \frac{1}{\exp \left[ \sum_{i=1}^{d} (n_i H(q_{h_i}) + m_i \ln n_i) + M \ln |Y| \right]}.
\]

**Lemma 9** For the prior defined in (35) and the posterior \( Q = \{ q(C_i|X_i) \}_{i=1}^{d} \):

\[
D(Q\|P) \leq \sum_{i=1}^{d} (n_i \bar{I}(X_i; C_i) + (m_i - 1) \ln n_i).
\]

And for the prior defined in (36) and the posterior \( Q = \{ \{ q(C_i|X_i) \}_{i=1}^{d}, q(Y|C_1, \ldots, C_d) \} \):

\[
D(Q\|P) \leq \sum_{i=1}^{d} (n_i \bar{I}(X_i; C_i) + m_i \ln n_i) + M \ln |Y|.
\]

#### 3.3.1 Proofs

**Proof of Lemma 8** To define the prior \( P \) over \( H_{\bar{m}} \) we count the hypotheses in \( H_{\bar{m}} \). There are \( \binom{n_i}{m_i - 1} \leq n_i^{m_i - 1} \) 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 \( q_{\bar{h}|i} = \{ |c_{i1}|, \ldots, |c_{im_i}| \} \) (over a single dimension) there are \((|c_{i1}|, \ldots, |c_{im_i}|) \) \( \leq e^{n_i H(q_{\bar{h}|i})} \) possibilities to assign \( X_i \)-s to the clusters. Putting the combinatorial calculations together we can define a distribution \( P(h) \) over \( \mathcal{H}_{\bar{m}} \) that satisfies (35).

To prove (36) we further define a uniform prior over \( \mathcal{H}_{|Y|\bar{m}} \). Note that there are \(|Y|^M \) 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 performed for (35) yields (36).

**Proof of Lemma 9**

We first handle the bound (37), where there are no labels. We use the decomposition \( D(Q\|P) = -E_Q P(h) - H(Q) \) and bound \(-E_Q P(h) \) and \( H(Q) \) separately. We further decompose \( P(h) = P(h_{|1}) \cdots P(h_{|d}) \) and \( Q(h) \) in a similar manner. Then \(-E_Q \ln P(h) = - \sum_i E_Q \ln P(h_{|i}) \), and similarly for \( H(Q) \). Therefore, we can treat each dimension separately.

To bound \(-E_Q \ln P(h_{|i}) \), recall that \( \bar{q}(c_i) = \frac{1}{n_i} \sum_{x_i} q(c_i|x_i) \) is the expected distribution over cardinalities of clusters along dimension \( i \) if we draw a cluster \( c_i \) for each value \( x_i \in X_i \) 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 8:

\[
-E_Q \ln P(h_{|i}) \leq (m_i - 1) \ln n_i + n_i \bar{E}_{q(c_i)} H(q_{h_{|i}}). 
\] (39)

To bound \( E_{q(c_i)} H(q_{h_{|i}}) \) we use the result on the negative bias of empirical entropy estimates cited below. See (Paninski, 2003) for a proof.

**Theorem 10 (Paninski, 2003)**

Let \( X_1, \ldots, X_N \) be i.i.d. distributed by \( p(X) \) and let \( \hat{p}(X) \) be their empirical distribution. Then:

\[
E_p H(\hat{p}) \leq H(p) - E_p D(\hat{p}\|p) 
\] (40)

By (40) \( E_{q(c_i)} H(q_{h_{|i}}) \leq H(\bar{q}(c_i)). \) Substituting this into (39) yields:

\[
-E_Q \ln P(h_{|i}) \leq n_i H(\bar{q}(c_i)) + (m_i - 1) \ln n_i. 
\] (41)

Now we turn to compute \(- H(Q) = E_Q \ln Q(h_{|i}) \). To do so we bound \( \ln Q(h_{|i}) \) 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 \( e^{-n_i H(C_i|X_i)} \), where \( H(C_i|X_i) = -\frac{1}{n_i} \sum_{x_i,c_i} q(c_i|x_i) \ln q(c_i|x_i) \) (see theorem 12.1.2 in (Cover and Thomas, 1991)). Thus, \( E_Q \ln Q(h_{|i}) \leq -n_i H(C_i|X_i) \), which together with (41) and the identity \( I(X_i;C_i) = H(\bar{q}(c_i)) - H(C_i|X_i) \) completes the proof of (37).

To prove (38) we recall that \( Q \) is defined for a fixed \( \bar{m} \). Hence, \(-E_Q \ln P(h_{|Y|\bar{m}}) = M \ln |Y| \) and \(- H(Q(h_{|Y|\bar{m}})) \leq 0 \). Finally, by the choice of prior \( P(\bar{m}) \) over the selection of \( \bar{m} \) we have \(-E_Q \ln P(\bar{m}) = \sum_{i=1}^d \ln n_i \) and \( H(Q(\bar{m})) = 0 \), which is added to (37) by the additivity of \( D(Q\|P) \) completing the proof.
3.4 PAC-Bayesian Analysis of Density Estimation with Grid Clustering

In this subsection 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 \(\mathcal{X}_1 \times \ldots \times \mathcal{X}_d\) based on a sample of size \(N\) from \(p\). As an illustrative example, think of estimating a joint probability distribution of words and documents \((X_1, X_2)\) from their co-occurrence matrix. The goodness of an estimator \(q\) for \(p\) is measured by \(\mathbb{E}_{\hat{p}(X_1, ..., X_d)} \ln q(X_1, ..., X_d)\).

By theorem 4, to obtain a meaningful bound for a direct estimation of \(p(X_1, ..., X_d)\) from \(\hat{p}(X_1, ..., X_d)\) we need \(N\) to be exponential in \(n_i\)-s, since the cardinality of the random variable \((X_1, ..., X_d)\) 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:

\[
q(X_1, ..., X_d) = \sum_{C_1, ..., C_d} q(C_1, ..., C_d) \prod_{i=1}^d q(X_i|C_i)
\]

\[= \sum_{C_1, ..., C_d} q(C_1, ..., C_d) \prod_{i=1}^d \frac{q(X_i)}{q(C_i)} q(C_i|X_i). \tag{42}
\]

We emphasize that the above decomposition assumption is only on the estimator \(q\) and not on the generating distribution \(p\).

We choose the hypothesis space \(\mathcal{H}\) to be the space of hard partitions of the product space \(\mathcal{X}_1 \times \ldots \times \mathcal{X}_d\), 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 \((C_1, ..., C_d)\) converges to the true one, and then we can use (42) to extrapolate it back on the whole space \(\mathcal{X}_1 \times \ldots \times \mathcal{X}_d\). Next we state this more formally.

We recall from the previous subsection that a distribution \(Q = \{q(C_i|X_i)\}_{i=1}^d\) is a distribution over \(\mathcal{H}_m\). To obtain a hypothesis \(h \in \mathcal{H}_m\) we draw a cluster for each \(x_i \in \mathcal{X}_i\) according to \(q(C_i|X_i)\). The way we have written (42) enables us to view it as a randomized prediction process: we draw a hypothesis \(h\) according to \(Q\) and then predict the probability of \((X_1, ..., X_d)\) as \(q(C^h_1(X_1), ..., C^h_d(X_d)) \prod_i q(X_i)/q(C^h_i(X_i))\), where \(C^h_i(X_i) = h(X_i)\) is the partition cell that \(X_i\) fell within in \(h\). Although (42) skips the process of drawing the complete partition \(h\) and returns the probability of \((X_1, ..., X_d)\) directly, the described randomized prediction process matches the predictions by (42) and thus enables us to apply the PAC-Bayesian bounds.

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

\[
p_h(c_1, ..., c_d) = \sum_{x_1, ..., x_d} p(x_1, ..., x_d) \quad \forall i, h(x_i) = c_i
\]

\[
p_h(c_i) = \sum_{x_i: h(x_i) = c_i} p(x_i). \tag{44}
\]

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:

$$\hat{p}_h(c_1, \ldots, c_d) = \sum_{x_1, \ldots, x_d; h(x_i) = c_i} \hat{p}(x_1, \ldots, x_d), \quad (45)$$

$$\hat{p}_h(c_i) = \sum_{x_i; h(x_i) = c_i} \hat{p}(x_i). \quad (46)$$

We also define the distribution over partition cells induced by $Q$ and $p$:

$$p_Q(c_1, \ldots, c_d) = \sum_h Q(h)p_h(c_1, \ldots, c_d) = \sum_{x_1, \ldots, x_d} p(x_1, \ldots, x_d) \prod_{i=1}^d q(c_i|x_i), \quad (47)$$

$$p_Q(c_i) = \sum_h Q(h)p_h(c_i) = \sum_{x_i} p(x_i)q(c_i|x_i). \quad (48)$$

And its empirical counterpart:

$$\hat{p}_Q(c_1, \ldots, c_d) = \sum_h Q(h)\hat{p}_h(c_1, \ldots, c_d) = \sum_{x_1, \ldots, x_d} \hat{p}(x_1, \ldots, x_d) \prod_{i=1}^d q(c_i|x_i), \quad (49)$$

$$\hat{p}_Q(c_i) = \sum_h Q(h)\hat{p}_h(c_i) = \sum_{x_i} \hat{p}(x_i)q(c_i|x_i). \quad (50)$$

We extrapolate $p_h$, $p_Q$, $\hat{p}_h$ and $\hat{p}_Q$ for the whole space $X_1 \times \ldots \times X_d$ using (42):

$$p_h(X_1, \ldots, X_d) = p_h(C_1^h(X_1), \ldots, C_d^h(X_d)) \prod_{i=1}^d \frac{p(X_i)}{p_h(C_i^h(X_i))}, \quad (51)$$

$$p_Q(X_1, \ldots, X_d) = \sum_{C_1, \ldots, C_d} p_Q(C_1, \ldots, C_d) \prod_{i=1}^d \frac{p(X_i)}{p_Q(C_i^h)} q(C_i|X_i), \quad (52)$$

$$\hat{p}_h(X_1, \ldots, X_d) = \hat{p}_h(C_1^h(X_1), \ldots, C_d^h(X_d)) \prod_{i=1}^d \frac{\hat{p}(X_i)}{\hat{p}_h(C_i^h(X_i))}, \quad (53)$$

$$\hat{p}_Q(X_1, \ldots, X_d) = \sum_{C_1, \ldots, C_d} \hat{p}_Q(C_1, \ldots, C_d) \prod_{i=1}^d \frac{\hat{p}(X_i)}{\hat{p}_Q(C_i^h)} q(C_i|X_i). \quad (54)$$

Note that $p_Q(X_1, \ldots, X_d)$ is the distribution over $X_1 \times \ldots \times X_d$, which has the form (42) and is the closest to the true distribution $p(X_1, \ldots, X_d)$ under the constraint that $\{q(C_i|X_i)\}_{i=1}^d$ are fixed. Further, note that since we have no access to $p(X_1, \ldots, X_d)$ we do not know $p_Q(X_1, \ldots, X_d)$. In the next theorem we state that the distributions $\hat{p}_Q(X_1, \ldots, X_d)$, $\hat{p}_Q(C_1, \ldots, C_d)$, and $\hat{p}_Q(X_i)$ based on the sample converge to their counterparts corresponding to the true distribution $p(X_1, \ldots, X_d)$.

**Theorem 11** For any probability measure $p$ over $X_1 \times \ldots \times X_d$ 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 $Q = \{q(C_i|X_i)\}_{i=1}^d$ the following holds simultaneously:

$$D(\hat{p}_Q(C_1, \ldots, C_d)\|p_Q(C_1, \ldots, C_d)) \leq \frac{\sum_{i=1}^d n_i \bar{I}(X_i; C_i) + K_1}{N} \quad (55)$$

21
and for all $i$

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

(56)

where

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

(57)

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

$$D(\hat{p}_Q(X_1, \ldots, X_d) \| p_Q(X_1, \ldots, X_d)) \leq \frac{\sum_{i=1}^{d} n_i \bar{I}(X_i; C_i) + K_2}{N},$$

(58)

where

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

(59)

Before we prove and discuss the theorem we point out that although $\hat{p}_Q(X_1, \ldots, X_d)$ converges to $p_Q(X_1, \ldots, X_d)$ it still cannot be used to minimize $-E_{p(X_1, \ldots, X_d)} \ln \hat{p}_Q(X_1, \ldots, X_d)$, because it is not bounded from zero. We also cannot construct a density estimator by smoothing $\hat{p}_Q(X_1, \ldots, X_d)$ directly using theorem 6, because the cardinality of the random variable $\langle X_1, \ldots, X_d \rangle$ is $\prod_i n_i$ and this term will enter into the bounds. To get around this we utilize the factor form of $p_Q$ and the bounds (55) and (56). We define an estimator $\tilde{p}_Q$, which is a smoothed version of $\hat{p}_Q$ in the following way:

$$\tilde{p}_h(C_1, \ldots, C_d) = \frac{\hat{p}_h(C_1, \ldots, C_d) + \gamma}{1 + \gamma M},$$

(60)

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

(61)

$$\tilde{p}_h(c_i) = \sum_{x_i : h(x_i) = c_i} \hat{p}(x_i),$$

(62)

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

(63)

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

$$\tilde{p}_Q(C_1, \ldots, C_d) = \frac{\hat{p}_Q(C_1, \ldots, C_d) + \gamma}{1 + \gamma M},$$

(64)

$$\tilde{p}_Q(C_i) = \sum_{x_i} \hat{p}(x_i) q(C_i | x_i) = \frac{\hat{p}_Q(C_i) + \gamma_i \bar{q}(C_i) n_i}{1 + \gamma_i n_i},$$

(65)

$$\tilde{p}_Q(X_1, \ldots, X_d) = \sum_{h} Q(h) \tilde{p}_h(X_1, \ldots, X_d)$$

$$= \sum_{C_1, \ldots, C_d} \hat{p}_Q(C_1, \ldots, C_d) \prod_{i=1}^{d} \frac{\tilde{p}(X_i)}{\hat{p}_Q(C_i)} q(C_i | X_i).$$

(66)

In the following theorem we provide a bound on $-E_{p(X_1, \ldots, X_d)} \ln \tilde{p}_Q(X_1, \ldots, X_d)$. Note that we take the expectation with respect to the true, unknown distribution $p$ that may have an arbitrary form.
Theorem 12 For the density estimator $\hat{p}_Q(X_1, \ldots, X_d)$ defined by equations (61), (64), (65), and (66), $-E_{p(X_1, \ldots, X_d)} \ln \hat{p}_Q(X_1, \ldots, X_d)$ attains its minimum at $\gamma = \frac{\sqrt{2}}{M}$ and $\gamma_i = \frac{\sqrt{2i}}{n_i}$, where $\varepsilon_i$ is defined by the right-hand side of (55) and $\varepsilon_i$ is defined by the right-hand side of (56). At this optimal level of smoothing, with a probability greater than $1 - \delta$ for all $Q = \{q(C_i|X_i)\}_{i=1}^d$ simultaneously:

$$-E_{p(X_1, \ldots, X_d)} \ln \hat{p}_Q(X_1, \ldots, X_d) \leq -I(\hat{p}_Q(C_1, \ldots, C_d)) + \ln(M) \left\{ \sum_{i=1}^d n_i I(X_i; C_i) + K_1 \right\} + 2N $$

where $I(\hat{p}_Q(C_1, \ldots, C_d)) = \left[ \sum_{i=1}^d H(\hat{p}_Q(C_i)) \right] - H(\hat{p}_Q(C_1, \ldots, C_d))$ is the multi-information between $C_1, \ldots, C_d$ with respect to $\hat{p}_Q(C_1, \ldots, C_d), K_1$ is defined by (57),

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

and the functions $\phi$ and $\psi$ are defined in theorem 6.

Discussion: We discuss theorem 12 first. We point out that $\hat{p}_Q(X_1, \ldots, X_d)$ is directly related to $\hat{p}(X_1, \ldots, X_d)$ and that $\hat{p}(X_1, \ldots, X_d)$ is determined by the empirical frequencies $\hat{p}(X_1, \ldots, X_d)$ of the sample and our choice of $Q = \{q(C_i|X_i)\}_{i=1}^d$. There are only two quantities in the bound (67) that depend on the choice of $Q$: $-I(\hat{p}_Q(C_1, \ldots, C_d))$ and $\sum_i \frac{n_i}{N} I(X_i; C_i)$ [note that the latter also appears in $\phi(\varepsilon)$]. Thus, theorem 12 suggests that a good estimator $\hat{p}_Q(X_1, \ldots, X_d)$ of $p(X_1, \ldots, X_d)$ should optimize the trade-off between $-I(\hat{p}_Q(C_1, \ldots, C_d))$ and $\sum_i \frac{n_i}{N} I(X_i; C_i)$. Similar to theorem 7, the latter term corresponds to the mutual information that the hidden cluster variables preserve on the observed variables. Larger values of $I(X_i; C_i)$ correspond to partitions of $X_1, \ldots, X_d$, which are more complex. The first term, $-I(\hat{p}_Q(C_1, \ldots, C_d))$, corresponds to the amount of structural information on $C_i$’s extracted by the partition. More precisely, we should look at the value of $\sum_i H(\hat{p}(X_i)) - I(\hat{p}_Q(C_1, \ldots, C_d))$, where $\sum_i H(\hat{p}(X_i))$ is a part of $K_3$ and roughly corresponds to the performance we can achieve by approximating $p(X_1, \ldots, X_d)$ with a product of empirical marginals $\hat{p}(X_1) \ldots \hat{p}(X_d)$. Thus, $-I(\hat{p}_Q(C_1, \ldots, C_d))$ is the added value of the partition in estimating $p(X_1, \ldots, X_d)$. Since $\sum_i H(\hat{p}(X_i)) \geq I(\hat{p}_Q(C_1, \ldots, C_d))$ the bound (67) is always positive.

The value of $I(\hat{p}_Q(C_1, \ldots, C_d))$ increases monotonically with the increase of the partition complexity $Q$ (we can see this by the information processing inequality (Cover and Thomas, 1991)). Thus, the trade-off in (67) is analogous to the trade-off in (31): the partition $Q$ should balance its utility function $-I(\hat{p}_Q(C_1, \ldots, C_d))$ and the statistical reliability of the estimate of the utility function, which is related to $\sum_i \frac{n_i}{N} I(X_i; C_i)$. This trade-off suggests a modification to the original objective of co-clustering in (Dhillon et al., 2003), which is maximization of $I(C_1; C_2)$ alone (Dhillon et al. (2003) discuss the case of two-dimensional matrices). The trade-off in (67) can be applied to model order selection.

Now we make a few comments concerning theorem 11. An interesting point about this theorem is that the cardinality of the random variable $(X_1, \ldots, X_d)$ is $\prod_i n_i$. Thus, a direct application of theorem 2 to bound $D(\hat{p}_Q(X_1, \ldots, X_d)||p_{Q}(X_1, \ldots, X_d))$ would introduce this term into the bound. However, by using the factor form (42) of $\hat{p}_Q(X_1, \ldots, X_d)$ and
$p_Q(X_1, \ldots, X_d)$ we are able to reduce this dependency to $(M + \sum_i n_i - d - 1)$. This result reveals a great potential of applying PAC-Bayesian analysis to more complex graphical models.

3.4.1 Proofs

We conclude this section by presenting the proofs of theorems 11 and 12.

**Proof of Theorem 11** The proof is based on the PAC-Bayesian theorem 2 on density estimation. To apply the theorem we need to define a prior $\mathcal{P}$ over $\mathcal{H}$ and then calculate $D(Q||P)$. We note that for a fixed $Q$ 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 11.

By theorem 2 and lemma 9, for the prior $\mathcal{P}$ over $\mathcal{H}_{\bar{m}}$ defined in lemma 8, with a probability greater than $1 - \frac{\delta}{(d+1)\prod_i n_i}$ we obtain (55) for each $\mathcal{H}_{\bar{m}}$. In addition, by theorem 4 with a probability greater than $1 - \frac{\delta}{d^N}$ inequality (56) 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 (55) and (56) hold simultaneously for all $Q$ and $X_i$ with a probability greater than $1 - \delta$.

To prove (58), fix some hard partition $h$ and let $C_i^h = h(X_i)$. Then:

$$D(\hat{p}_h(X_1, \ldots, X_d) || p_h(X_1, \ldots, X_d))$$

$$= D(\hat{p}_h(X_1, \ldots, X_d, C_1^h(X_1), \ldots, C_d^h(X_d)) || p_h(X_1, \ldots, X_d, C_1^h(X_1), \ldots, C_d^h(X_d)))$$

$$= D(\hat{p}_h(C_1, \ldots, C_d) || p_h(C_1, \ldots, C_d))$$

$$+ D(\hat{p}_h(X_1, \ldots, X_d | C_1^h(X_1), \ldots, C_d^h(X_d)) || p_h(X_1, \ldots, X_d | C_1^h(X_1), \ldots, C_d^h(X_d)))$$

$$= D(\hat{p}_h(C_1, \ldots, C_d) || p_h(C_1, \ldots, C_d)) + \sum_{i=1}^d D(\hat{p}_h(X_i | C_i^h(X_i)) || p_h(X_i | C_i^h(X_i)))$$

$$= D(\hat{p}_h(C_1, \ldots, C_d) || p_h(C_1, \ldots, C_d)) + \sum_{i=1}^d D(\hat{p}(X_i) || p(X_i)) - \sum_{i=1}^d D(\hat{p}(C_i) || p(C_i))$$

$$\leq D(\hat{p}_h(C_1, \ldots, C_d) || p_h(C_1, \ldots, C_d)) + \sum_{i=1}^d D(\hat{p}(X_i) || p(X_i)).$$

And:

$$E_S \mathbb{E}^{ND(\hat{p}_h(X_1, \ldots, X_d) || p_h(X_1, \ldots, X_d))} \leq \left( E_S \mathbb{E}^{ND(\hat{p}_h(C_1, \ldots, C_d) || p_h(C_1, \ldots, C_d))} \right)^{d} \prod_{i=1}^d E_S \mathbb{E}^{ND(\hat{p}(X_i) || p(X_i))}$$

$$\leq (N + 1)^{M + \sum_{i=1}^d n_i - (d+1)},$$

where the last inequality is by theorem 3. From here we follow the lines of the proof of theorem 2. Namely:

$$E_S \left[ E_{\mathcal{P}(\hat{h})}^{ND(\hat{p}_h(X_1, \ldots, X_d) || p_h(X_1, \ldots, X_d))} \right] = E_{\mathcal{P}(\hat{h})} \left[ E_S \mathbb{E}^{ND(\hat{p}_h(X_1, \ldots, X_d) || p_h(X_1, \ldots, X_d))} \right]$$

$$\leq (N + 1)^{M + \sum_{i=1}^d n_i - (d+1)}.$$
Thus, by Markov’s inequality
\[ \mathbb{E} \mathcal{P}(\hat{h}(X_1, ..., X_d) | p_h(X_1, ..., X_d)) \leq \frac{1}{\delta} (N + 1)^{M + \sum_i n_i} (d + 1) \]
with a probability of at least \( 1 - \delta \) and (58) follows by the change of measure inequality
(15) and convexity of the KL-divergence, when the prior \( \mathcal{P} \) over \( \mathcal{H} \) defined in lemma 8 is selected (this time we give a weight of \( (\prod_i n_i)^{-1} \) to each \( \mathcal{H}_m \) and obtain a prior over the whole \( \mathcal{H} \)). The calculation of \( D(Q||\mathcal{P}) \) for this prior is done in lemma 9.

**Proof of Theorem 12**

\[
- \mathbb{E}_{p(X_1, ..., X_d)} \ln \tilde{p}_Q(X_1, ..., X_d) = - \mathbb{E}_{p(X_1, ..., X_d)} \ln \mathbb{E}_{Q(h)} \tilde{p}_h(X_1, ..., X_d)
\leq - \mathbb{E}_{Q(h)} \mathbb{E}_{p(X_1, ..., X_d)} \ln \tilde{p}_h(X_1, ..., X_d)
= - \mathbb{E}_{Q(h)} \mathbb{E}_{p(X_1, ..., X_d)} \ln \tilde{p}_h(C_1(X_1), ..., C_d(X_d)) \prod_i \frac{\tilde{p}(X_i)}{\tilde{p}_h(C_i(X_i))}
= - \mathbb{E}_{Q(h)} [\mathbb{E}_{p(C_1, ..., C_d)} \ln \tilde{p}_h(C_1, ..., C_d)] - \sum_i \mathbb{E}_{p(X_i)} \ln \tilde{p}(X_i) + \sum_i \mathbb{E}_{Q(h)} \mathbb{E}_{p(Q(C_i))} \ln \tilde{p}_Q(C_i)
\leq - \mathbb{E}_{Q(h)} [\mathbb{E}_{p(C_1, ..., C_d)} \ln \tilde{p}_h(C_1, ..., C_d)] - \sum_i \mathbb{E}_{p(X_i)} \ln \tilde{p}(X_i) + \sum_i \mathbb{E}_{p(C_i)} \ln \tilde{p}_Q(C_i)
\]

At this point we use (23) to bound the first and the second term and the lower bound (25) to bound the last term and obtain (67).

4. **PAC-Bayesian Analysis of Graphical Models**

The analysis of co-clustering presented in the previous section holds for any dimension \( d \). However, the dependence of the bounds (31), (58), and (67) on \( d \) is exponential because of the \( M = \prod_i m_i \) term they involve. This term is reasonably small when the number of dimensions is small (two or three), as in the example of co-clustering. But 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.

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 \( d = 1 \)). However, the one-dimensional case falls out of the main discussion line of this paper and we refer the interested reader to (Seldin and Tishby, 2008; Seldin, 2009) for further details.

In this section we suggest a hierarchical approach to handle high-dimensional problems. Then we show that this approach can also be applied to generalization analysis of graphical models.

4.1 **Hierarchical Approach to High Dimensional Problems (\( d > 2 \))**

One possible way to handle high dimensional problems is to use hierarchical partitions, as shown in Figure 4. For example, the discriminative prediction rule corresponding to the
Seldin and Tishby

(a) Discriminative Prediction

(b) Density Estimation

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

model in Figure 4.a is:

$$q(Y|X_1, ..., X_4) = \sum_{D_1, D_2} q(Y|D_1, D_2) \sum_{C_1, ..., C_4} \prod_{i=1}^{2} q(D_i|C_{2i-1}, C_{2i}) \prod_{j=1}^{4} q(C_j|X_j).$$  (68)

And the corresponding randomized prediction strategy is

$$\mathcal{Q} = \{\{q(C_i|X_i)\}_{i=1}^{4}, \{q(D_i|C_{2i-1}, C_{2i})\}_{i=1}^{2}, q(Y|D_1, D_2)\}. \text{ In this case the hypothesis space is the space of all hard partitions of } X_i \text{-s to } C_i \text{-s and of the pairs } (C_{2i-1}, C_{2i}) \text{ to } D_i \text{-s. By repeating the analysis in theorem 7 we obtain that with a probability greater than } 1 - \delta:}$$

$$D_b(\hat{L}(\mathcal{Q})||L(\mathcal{Q})) \leq \frac{F_1 + F_2 + |D_1||D_2|\ln|Y| + \frac{1}{2}\ln(4N) - \ln\delta}{N},$$  (69)

where

$$F_1 = \sum_{i=1}^{4} \left( n_i \hat{I}(X_i; C_i) + m_i \ln n_i \right),$$  (70)

$$F_2 = \sum_{i=1}^{2} \left( (m_{2i-1}m_{2i}) \hat{I}(D_i; C_{2i-1}, C_{2i}) + |D_i| \ln(m_{2i-1}m_{2i}) \right).$$  (71)

Observe that the $M \ln|Y|$ term in (31), which corresponds to the clique $\langle C_1, C_2, C_3, C_4, Y \rangle$, is replaced in (69) 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 4. Imagine that 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. The subspace of pairs of amino acids is only $20^2 = 400$ instances and local interactions between adjacent pairs of amino acids can easily be studied. We can cluster the pairs of amino acids into, 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. However, the reduced subspace of pairs of $C_i$-s is only $20^2 = 400$ instances. 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 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 (69) suggests how this trade-off between model resolution and statistical reliability can be optimized. It is further possible to derive analogs to (58) and (67) that apply to density estimation hierarchies (as in Figure 4.b) in a similar manner.

4.2 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 (68) and bound (69) both correspond to the undirected graph in Figure 4.a and to the directed graph in Figure 6. (In fact, Figure 4.a is a moralized counterpart of the directed acyclic graph in Figure 6 (Cowell et al., 2007).)

The analysis used to derive bound (69) can be applied to any directed graphical model in the form of a tree (directed up, as in Figure 6) or its moralized counterpart. The analysis shows that the generalization power of these graphical models is determined by a trade-off between empirical performance and the amount of information that is propagated up the
Figure 6: **Illustration of a directed graphical model.** A corresponding moral graph is depicted in Figure 4.a.

It is important to note that the PAC-Bayesian bound is able to utilize the factor form of distribution (68) and that bound (69) depends on the sizes of the tree cliques, but not on the size of the parameter space $X_1 \times \ldots \times X_d$. 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 are key directions for future research.

### 5. Algorithms

In section 3 we presented generalization bounds for discriminative prediction and density estimation with co-clustering. The bounds presented in theorems 7 and 12 hold for any prediction rule $Q$ based on grid clustering of the parameter space $X_1 \times \ldots \times X_d$. In this section we address the question of how to find local minima of the bounds. In (Seldin, 2009) it is shown that for $d = 1$ the global minimum can be found efficiently. However, for $d \geq 2$ it is exponentially hard to find the global minimum.

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 section we suggest how to replace the bounds with a trade-off that can be further fine tuned, e.g., via cross-validation, to improve their usability in practice.

#### 5.1 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 (31) suggested in theorem 7. For convenience we quote the bound (31) below once again:

$$D_b(\tilde{L}(Q)||L(Q)) \leq \frac{\sum_{i=1}^{d} (n_i \tilde{I}(X_i; C_i) + m_i \ln n_i) + M \ln |Y| + \frac{1}{2} \ln (4N) - \ln \delta}{N}.$$
We further rewrite it in a slightly different way:

$$D_b(\hat{L}(Q)||L(Q)) \leq \frac{\sum_{i=1}^{d} n_i \bar{I}(X_i; C_i) + K}{N},$$  \hspace{1cm} (72)

where

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

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 selected, $K$ is constant.

The minimization problem corresponding to (72) can be stated as follows:

$$\min_{Q} L \quad s.t. \quad D_b(\hat{L}(Q)||L) = \frac{\sum_{i=1}^{d} n_i \bar{I}(X_i; C_i) + K}{N}. \hspace{1cm} (73)$$

It is generally possible to find a local minimum of the minimization problem (73) directly using alternating projection methods - see, e.g., (Germain et al., 2009) 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, $L(Q)$ depends on the trade-off between $\hat{L}(Q)$ and $\sum_{i=1}^{d} n_i \bar{I}(X_i; C_i)$, which can be written as follows:

$$\mathcal{F}(Q) = \beta N \hat{L}(Q) + \sum_{i=1}^{d} n_i \bar{I}(X_i; C_i). \hspace{1cm} (74)$$

The minimization problem (73) is then replaced by:

$$\min_{Q} \beta N \hat{L}(Q) + \sum_{i=1}^{d} n_i \bar{I}(X_i; C_i). \hspace{1cm} (75)$$

In general, every value of $\beta$ yields a different solution to the minimization problem (75). The optimum of (73) (which is computationally hard to find) corresponds to some specific value of $\beta$. Hence, by scanning the possible values of $\beta$ and minimizing (75) it is virtually possible to find the optimum of (73) (only virtually, because finding the global optimum of (75) is computationally hard as well). However, the trade-off (74) provides us an additional degree of freedom. In cases where the bound (31) is not sufficiently tight for practical applications it is possible to tune the trade-off by determining the desired value of $\beta$ via cross-validation instead of back-substitution into the bound.

The minimization problem (75) is closely related to the rate distortion trade-off in information theory (Cover and Thomas, 1991). To find a local minimum of $\mathcal{F}(Q)$ we adapt an EM-like alternating projection procedure, very similar to the Blahut-Arimoto algorithm for minimization of the rate distortion function (Arimoto, 1972; Blahut, 1972; Cover and Thomas, 1991). We note that for $d \geq 2$ the alternating projection involves more than two convex sets and hence only a local minimum can be achieved. (For $d = 1$ the procedure achieves the global minimum.) For the sake of simplicity of the notations we restrict ourselves to the case of $d = 2$. 
The Lagrangian corresponding to the minimization problem (75) is:

\[ \mathcal{L}(Q) = \beta N \hat{L}(Q) + \sum_{i=1}^{2} n_i \tilde{I}(X_i; C_i) + \sum_{i=1}^{2} \sum_{x_i \in X_i} \nu(x_i) \sum_{c_i} q(c_i|x_i) + \sum_{c_1, c_2} \nu(c_1, c_2) \sum_{y} q(y|c_1, c_2), \]

where \( \nu \)-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 \( \mathcal{L}(Q) \) we write \( \hat{L}(Q) \) explicitly:

\[
\begin{align*}
\hat{L}(Q) &= \sum_{x_1, x_2, y} \tilde{p}(x_1, x_2, y) \sum_{y'} q(y'|x_1, x_2) l(y, y') \\
&= \sum_{x_1, x_2, y} \tilde{p}(x_1, x_2, y) \sum_{y', c_1, c_2} q(y'|c_1, c_2) q(c_1|x_1) q(c_2|x_2) l(y, y') \\
&= \sum_{y, y'} l(y, y') \sum_{c_1, c_2} q(y'|c_1, c_2) \sum_{x_1, x_2} q(c_1|x_1) \tilde{p}(x_1, x_2, y) q(c_2|x_2).
\end{align*}
\]

We further derive \( \hat{L}(Q) \) 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.

\[
\frac{\partial \hat{L}(Q)}{\partial q(C_1|X_1)} = \sum_{y, y'} l(y, y') \sum_{x_2, c_2} \tilde{q}(y'|C_1, c_2) p(X_1, x_2, y) q(c_2|x_2). \tag{77}
\]

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 \( \mathcal{L}(Q) \) provide normalization for the corresponding variables. Thus, by taking the derivative of \( \mathcal{L}(Q) \) 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:

\[
\begin{align*}
\tilde{q}_t(c_i) &= \frac{1}{n_i} \sum_{x_i} q_t(c_i|x_i) \tag{78} \\
q_{t+1}(c_i|x_i) &= \frac{\tilde{q}_t(c_i)}{Z_{t+1}(x_i)} e^{-\beta N \frac{\partial L(Q)}{\partial q(C_i|X_i)}} \tag{79} \\
Z_{t+1}(x_i) &= \sum_{c_i} q_{t+1}(c_i|x_i) \tag{80}
\end{align*}
\]
\[ y_{t+1}^*(c_1, c_2) = \arg\min_{y} \sum_{y'} l(y, y') \sum_{x_1, x_2} q_{t+1}(c_1|x_1) \hat{p}(x_1, x_2, y) q_{t+1}(c_2|x_2), \]  
\[ q_{t+1}(y|c_1, c_2) = \delta[y, y_{t+1}^*(c_1, c_2)], \]

where $\delta[\cdot, \cdot]$ is the Kronecker delta function, $\frac{\partial \hat{L}(Q)}{\partial q(C_i|X_i)}$ is given by (77), and the subindex $t$ denotes the iteration number. Equations (82) and (81) correspond to minimization of $\hat{L}(Q)$ 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 $(c_1, c_2)$ partition cell; for the absolute loss it is the median value; for the quadratic loss it is the average value. We summarize this in the Algorithm 1 box. We note that for 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.

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

**Input:** $\hat{p}(x_1, x_2, y)$, $N$, $n_1$, $n_2$, $m_1$, $m_2$, $l(y, y')$, $|Y|$, $\beta$.

**Initialize:** $q_0(C_i|X_i)$ and $q_0(Y|C_1, C_2)$ randomly.

$t \leftarrow 0$

$q_0(c_i) \leftarrow \frac{1}{n_i} \sum_{x_i} q_t(c_i|x_i)$

**repeat**

for $i = 1, 2$

$q_{t+1}(c_i|x_i) \leftarrow \frac{\hat{q}_t(c_i) e^{-\beta N \frac{\partial \hat{L}(Q_t)}{\partial q(c_i|x_i)}}}{\sum_{c_i} q_{t+1}(c_i|x_i)}$

$Z_{t+1}(x_i) \leftarrow \sum_{c_i} q_{t+1}(c_i|x_i)$

$q_{t+1}(c_i|x_i) \leftarrow \frac{q_{t+1}(c_i|x_i)}{Z_{t+1}(x_i)}$

$q_{t+1}(c_i) \leftarrow \frac{1}{n_i} \sum_{x_i} q_{t+1}(c_i|x_i)$

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

$q_{t+1}(y|c_1, c_2) \leftarrow \delta[y, y_{t+1}^*(c_1, c_2)]$

$t \leftarrow t + 1$

**end for**

**until** convergence

**return** $\{q_t(C_i|X_i)\}_{i=1}^{2}, q_t(Y|C_1, C_2)$ from the last iteration.

5.2 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 (67) depends on the trade-off:

\[ \mathcal{G}(Q) = -\beta NI(\hat{p}_Q(C_1, C_2)) + \sum_{i=1}^{2} n_i \hat{I}(X_i; C_i). \]

All other terms in (67) do not depend on the specific form of grid partition $Q$. (As in the previous section we restrict ourselves to $d = 2$.) Unfortunately, $-I(\hat{p}_Q(C_1, C_2))$ is concave
in $q(C_i|X_i)$-s, whereas $\bar{I}(X_i; C_i)$ is convex in $q(C_i|X_i)$. Therefore, alternating projection methods are hard to apply. Instead, $\mathcal{G}(Q)$ can be minimized using sequential minimization (Slonim et al., 2001; Dhillon et al., 2003). 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 $\mathcal{G}(Q)$ 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 Algorithm 2 box.

<table>
<thead>
<tr>
<th>Algorithm 2 Algorithm for sequential minimization of $\mathcal{G}(Q) = -\beta N I(\hat{p}<em>Q(C_1, C_2)) + \sum</em>{i=1}^{2} n_i \bar{I}(X_i; C_i)$.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $\hat{p}(x_1, x_2)$, $N$, $n_1$, $n_2$, $m_1$, $m_2$, $\beta$.</td>
</tr>
<tr>
<td><strong>Initialize:</strong> $q_0(C_i</td>
</tr>
<tr>
<td><strong>repeat</strong></td>
</tr>
<tr>
<td><strong>for</strong> all $x_1 \in X_1$ and all $x_2 \in X_2$ according to some random order over $X_1$ and $X_2$ <strong>do</strong></td>
</tr>
<tr>
<td><strong>for</strong> $i = 1, 2$ <strong>do</strong></td>
</tr>
<tr>
<td>Select $x_i \in X_i$ according to the order selected above.</td>
</tr>
<tr>
<td>Compute $\mathcal{G}(Q)$ for each possible assignment of $x_i$ to $c_i \in {1, \ldots, m_i}$</td>
</tr>
<tr>
<td>Reassign $x_i$ to $c_i$ such that $\mathcal{G}(Q)$ is minimized.</td>
</tr>
<tr>
<td>Update $\hat{p}<em>Q(C_1, C_2) \leftarrow \sum</em>{x_1, x_2} q(C_1</td>
</tr>
<tr>
<td><strong>end for</strong></td>
</tr>
<tr>
<td><strong>end for</strong></td>
</tr>
<tr>
<td><strong>until</strong> no reassignments further minimize $\mathcal{G}(Q)$.</td>
</tr>
<tr>
<td><strong>return</strong> ${q(C_i</td>
</tr>
</tbody>
</table>

6. Applications

In this section we illustrate an application of the PAC-Bayesian bound for discriminative prediction based on co-clustering (31) and the Algorithm 1 for its minimization on the problem of collaborative filtering. The problem of collaborative filtering was discussed in the previous sections. 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 challenge2. 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 dataset3. 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) measure, which is commonly used for evaluation on this dataset (Herlocker et al., 2004). Let $\hat{p}(x_1, x_2, y)$ be the distribution over $(X_1, X_2, Y)$ in the test set. The mean absolute error is

---

defined as:

\[
MAE = \sum_{x_1, x_2, y} \bar{p}(x_1, x_2, y) \sum_{y'} q(y'|x_1, x_2) |y - y'|.
\]  

(84)

In previous work the best MAE reported for this dataset was 0.73 (Herlocker et al., 2004). 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 (Seldin et al., 2007) we improved the MAE on this dataset to 0.72 by using an 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 trade-off (74) for \(\beta = 1\). For convenience we cite (74) below once again:

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

In the MDL formulation of co-clustering developed in (Seldin et al., 2007) 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 1 for minimization of \(\mathcal{F}(Q)\) (for an arbitrary value of \(\beta\)) and applied it to the MovieLens dataset. There are four major differences between the Algorithm 1 and the algorithm for minimization of \(\mathcal{F}(Q)\) suggested in (Seldin et al., 2007) that should be highlighted:

- Algorithm 1 considers soft assignments of \(X_i\)-s to \(C_i\)-s.
- Algorithm 1 is an iterative projection algorithm rather than the sequential optimization algorithm suggested in (Seldin et al., 2007). Note that this point is neither positive nor negative, since sequential optimization algorithms are very powerful and especially in hard cases can outperform iterative projection methods. The advantage of iterative projection 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 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 1 considers arbitrary values of \(\beta\). (However, the algorithm in (Seldin et al., 2007) can be easily extended to handle arbitrary values of \(\beta\).) As we will show below, \(\beta = 1\) is not always optimal.
We conducted three experiments with Algorithm 1. In all three experiments we fixed the numbers of clusters $m_1$ and $m_2$ used along both dimensions and analyzed the MAE on the test set and the value of the bound (31) as a function of $\beta$. In each experiment, for each of the five splits of the dataset into train and test sets mentioned earlier, and for each value of $\beta$ we applied 10 random initializations of the algorithm. The solution $Q$ corresponding to the best value of $F(Q)$ per each data split and per each value of $\beta$ was then selected. We further calculated the average of the results over the dataset splits to produce the graphs of the bound values and test MAE as functions of $\beta$.

In the first experiment we verified that we are able to reproduce the results achieved previously in (Seldin et al., 2007). We set $m_1 = 13$ and $m_2 = 6$, as the best values obtained in (Seldin et al., 2007) and applied Algorithm 1. The results are presented in Figure 7. We make the following conclusions from this experiment:

- The performance of Algorithm 1 is comparable to the performance achieved in (Seldin et al., 2007) with sequential optimization.
- The optimal performance is achieved at $\beta$ close to one, which corresponds to the MDL functional optimized in (Seldin et al., 2007).
- The values of the bound are meaningful (recall that the maximal possible loss is 4; thus the bound value of $\sim 1.25$ 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 explained by

Figure 7: **Co-clustering of the MovieLens dataset into 13x6 clusters.** Figure a. shows the value of bound (31) together with the MAE on the test set as a function of $\beta$. Figure b. zooms into MAE on the test set. The values of $\beta$ are on a log scale. See text for further details.
Figure 8: Co-clustering of the MovieLens dataset into 50x50 clusters. Figure a. shows the value of bound (31) together with the MAE on the test set as a function of $\beta$. Figure b. zooms into MAE on the test set. The values of $\beta$ are on a log scale. See text for further details.

the fact that this is a hard problem and the improvement in the empirical loss $\hat{L}(Q)$ achieved by co-clustering is relatively small. For the best co-clustering solution found $\hat{L}(Q) \approx 0.67$, whereas if we assign all the data to one big cluster $\hat{L}(Q) \approx 0.89$. Thus, the improvement in $\hat{L}(Q)$ achieved by the clustering is only about 30% while 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 $\beta$ in the trade-off $F(Q)$ via cross-validation and obtain remarkably good results. It should be pointed out that the trade-off $F(Q)$ 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 $\beta$ beyond 1. This will change in the following experiments.

The power of bound (31) and the trade-off $F(Q)$ derived from the bound is that it 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 said available clusters. This property is verified in the following two experiments. In the first of them we initialized Algorithm 1 with $m_1 = m_2 = 50$ clusters along each dimension. The result of optimization of $F(Q)$ as a function of $\beta$ is presented in Figure 8. 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
Figure 9: Co-clustering of the MovieLens dataset into 283x283 clusters. Figure a. shows the value of bound (31) together with the MAE on the test set as a function of $\beta$. Figure b. zooms into MAE on the test set. The values of $\beta$ are on a log scale. See text for further details.

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 $\beta$ 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 $\beta$ 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 would still prefer to set the value of $\beta$ 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 $M = m_1 m_2$ in this case is 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 trade-off parameter $\beta$. The result of the experiment is presented in Figure 9. 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 over regularization of the model via parameter $\beta$ of the trade-off $\mathcal{F}(Q)$. 

The role of the regularization parameter $\beta$ is further increased in this experiment compared to the previous two. The optimal value of $\beta$ 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 $\beta$ should be set via cross-validation.

7. Stochastic-Form Matrix Factorization

Note that for $d = 2$ the prediction model based on co-clustering

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

can be considered as a form of matrix factorization

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

(In (85) each partition cell $\langle C_1, C_2 \rangle$ is assigned a deterministic label $f(C_1, C_2)$.) More specifically, let $A$ be an $n_1 \times n_2$ matrix, possibly sparse, with the values of $Y$ observed for corresponding combinations of $(X_1, X_2)$. (Recall that $n_i = |X_i|$.) For example, $A$ can be a viewers-by-movies collaborative filtering matrix holding the ratings. Then we can write:

$$A \approx L^T MR,$$

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. This form of matrix factorization was already considered in (Banerjee et al., 2007). However, in (Banerjee et al., 2007) the matrices $L$ and $R$ are restricted to deterministic assignments of $X_i$-s to $C_i$-s (the entries of $L$ and $R$ are in $\{0, 1\}$), whereas in the factorization proposed here $L$ and $R$ are stochastic matrices and $M$ is arbitrary. Therefore, we term it a Stochastic-Form Matrix Factorization (SFMF). Algorithm 1 can be applied to find a locally optimal factorization.

There are several advantages to SFMF:

- It has a clear probabilistic interpretation.
- It naturally handles missing values.
• Overfitting can be controlled via the regularization parameter $\beta$.

• 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 in advance, but this requirement can be relaxed.)

• As we have shown in the previous section, SFMF achieves state-of-the-art prediction performance on collaborative filtering of the MovieLens dataset.

SFMF can be compared with other matrix factorization forms including singular value decomposition (SVD) (Strang, 2009; Golub and Loan, 1996), non-negative matrix factorization (Lee and Seung, 1999, 2001), low-rank matrix factorization (Srebro et al., 2005a), and maximum margin matrix factorization (Srebro et al., 2005b; Srebro, 2004). The clear probabilistic interpretation is an advantage of SFMF over all the approaches mentioned. The generalization analysis involving mutual information terms also seems more appropriate for combinatorial problems, such as matrix factorization, as opposed to $L_2$-norm regularization in maximum margin factorization, which is more appropriate for continuous problems. As we have shown in the previous section, SFMF achieves state-of-the-art prediction performance on collaborative filtering of the MovieLens dataset. We leave a wider practical comparison of the different matrix factorization methods as a subject for future work.

Another promising direction for future research is to apply SFMF in tasks where multiple related datasets are considered. For example, let $A_1$ be a collaborative filtering matrix, $A_2$ be a matrix of viewers-by-viewers properties and $A_3$ be a matrix of movies-by-movies properties. We can look for simultaneous factorizations such that:

$$A_1 \approx L_1^T M_1 R_1$$
$$A_2 \approx L_1^T M_2 R_2$$
$$A_3 \approx L_3^T M_3 R_1.$$

In other words, the clustering of viewers into clusters of viewers is shared between factorization of $A_1$ and $A_2$ and clustering of movies into clusters of movies is shared between factorization of $A_1$ and $A_3$.

The above case is frequent in bioinformatics, when multiple experiments with partial relations are considered. For example, Alter et al. (2003) 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, SFMF has several advantages over SVD (and consequently over GSVD). Hence, it would be interesting to apply it to this type of problem.

8. Related Work

The idea of considering clustering in the context of a higher level task was inspired by the Information Bottleneck (IB) principle (Tishby et al., 1999; Slonim, 2002; Slonim et al., 2006). The IB principle considers the problem of extracting information from a random variable $X$ that is relevant for the prediction of a random variable $Y$. The relevance variable
Y defines the high-level task. The extraction of relevant information from X is done by means of clustering of X into clusters \( \hat{X} \) (Tishby et al., 1999). In other words, IB is looking for the structure \( \hat{X} \) of X that is relevant for the prediction of Y. The IB principle was further extended to graphical models in (Slonim et al., 2006). In the example with the cubes given in the introduction we can define \( X_1, \ldots, X_d \) to be various parameters of the cubes, where \( d \) is the number of the parameters considered for each cube. For example, \( X_1 \) can be the shape of the cube, \( X_2 \) cube's color, \( X_3 \) weight, etc. The relevance variable \( Y \) may be an indicator variable as to whether the cube fits into a certain volume or a real-valued variable, for example the coefficient of light energy absorption. Clearly, each relevance variable corresponds to a different partition (clustering) of the parameter space.

The idea to consider clustering as a proxy to a solution of a prediction task was further developed in (Krupka and Tishby, 2005, 2008; Krupka, 2008). In these works Krupka and Tishby analyze a scenario wherein each object has multiple properties, but only a fraction of the properties is observed. Consider the following illustration: assume we are presented with multiple fruits and we observe their parameters, such as size, color, and weight. We can cluster the fruits by their observed parameters in order to facilitate prediction of unobserved parameters, such as taste and toxicity. This approach enables one to conduct a formal analysis and derive generalization bounds for prediction rules based on clustering.

In recent years extensive attempts have been made to address the question of model order selection in clustering through evaluation of its stability (Lange et al., 2004; von Luxburg and Ben-David, 2005; Ben-David et al., 2006; Shamir and Tishby, 2009; Ben-David and von Luxburg, 2008). This perspective suggests that for two random samples generated by the same source, clustering of the samples should be similar (and hence stable). Otherwise the obtained clustering is unreliable. Although it has been proved that in a large sample regime stability can be used for model order selection (Shamir and Tishby, 2009), no upper bounds on the minimal sample size required for stability estimates to hold can be proved. Furthermore, in certain cases stability indices based on arbitrarily large samples can be misleading (Ben-David and von Luxburg, 2008). Since in any practical application the amount of data available is limited, currently existing stability indices cannot be used for reliable model order selection. Moreover, it is not clear whether the stability indices can be used to compare solutions based on different optimization objectives.

**Gaussian ring example.** We use the following example from (Seldin, 2009) to illustrate that generalization and stability criteria for evaluation of clustering are not equivalent. Assume points in \( \mathbb{R}^2 \) are generated according to the following process. First, we select a center \( \mu \) 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 \( \mu \) with a covariance matrix \( \sigma^2 I \) for a fixed \( \sigma \) (\( I \) is a 2 by 2 identity matrix). 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:

1. 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).

2. 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 clustering.

9. Discussion and Future Work

This paper underlines the importance of external evaluation of unsupervised learning, such as clustering or more general structure learning, based on the context of its potential application. Such form of evaluation is important for delivery of better structure learning algorithms as well as for better understanding of their outcome. We argue that structure learning does not occur for its own sake, but rather in order to facilitate a solution of some higher level goal. In any non-trivial data many structures co-exist simultaneously and it is a matter of the subsequent usage of the outcome of the learning algorithm to determine which of the structure elements are valuable to be found and which are not. Therefore, unsupervised learning cannot be analyzed in isolation of its potential application. In our opinion, one of the main obstacles in theoretical advancement of unsupervised learning is an absence of a good mathematical definition of the context of its application. The analysis of co-clustering presented here is a first step toward context-based analysis of more complex models.

The work presented here started with an attempt to improve our understanding of clustering. We note that clustering is tightly related to object naming process in the human language. In a sense, a cluster is an entity that can be assigned a name. By clustering objects we ignore their irrelevant properties and concentrate on the relevant ones. And of course, this division can change according to our needs. For example, we can divide animals into birds and mammals or into flying and notatorial or into domestic and wild. Whereas the classification into birds and mammals or flying and notatorial may be considered intrinsic, the classification into domestic and wild is definitely application-oriented. In order to design successful clustering and categorization algorithms it is important to understand the basic principles behind this process. It is not a-priori clear that, if we restrict ourselves to pure prediction tasks, clustering the underlying sample space helps. As shown in (Seldin and Tishby, 2008; Seldin, 2009), in classification by a single parameter there is no need to cluster the parameter space, but rather simple smoothing performs better. In classification in higher dimensional spaces, kernel-based methods can be superior to clustering-based approaches. However, we know that as humans we communicate by using a clustered representation of the world rather than by kernel matrices. Thus, there should be advantages for such form of communication. Identification, understanding, and analysis of these advantages is an important future direction for the design of better clustering and higher structure learning algorithms.

Appendix A. Proof of theorem 6

Proof of theorem 6 First we prove (23):

\[-E_{Q(h)}E_{p_h(Z)} \ln \tilde{p}_h(Z) = E_{Q(h)}E_{[\tilde{p}_h(Z) - p_h(Z)]} \ln \tilde{p}_h(Z) - E_{Q(h)}E_{\tilde{p}_h(Z)} \ln \tilde{p}_h(Z)\]

\[= E_{Q(h)}E_{[\tilde{p}_h(Z) - p_h(Z)]} \ln \frac{\tilde{p}_h(Z) + \gamma}{1 + \gamma |Z|} - E_{Q(h)}E_{\tilde{p}_h(Z)} \ln \frac{\tilde{p}_h(Z) + \gamma}{1 + \gamma |Z|}\]
\[
\leq -\frac{1}{2} \|\hat{p}_Q(Z) - p_Q(Z)\|_1 \ln \frac{1 + \gamma|Z|}{\gamma} + \mathbb{E}_{Q(h)} H(\hat{p}_h(Z)) + \ln(1 + \gamma|Z|)
\]
\[
\leq H(\hat{p}_Q(Z)) - \sqrt{\varepsilon/2} \ln \frac{1 + \gamma|Z|}{\gamma} + \ln(1 + \gamma|Z|). \tag{87}
\]

The last inequality is justified by the concavity of the entropy function \(H\) and the KL-divergence bound on the \(L_1\) norm (Cover and Thomas, 1991):
\[
\|\hat{p}_Q(Z) - p_Q(Z)\|_1 \leq \sqrt{2D(\hat{p}_Q(Z)\|p_Q(Z))} \leq \sqrt{2\varepsilon}. \tag{88}
\]

By differentiation (87) is minimized by \(\gamma = \frac{\sqrt{\varepsilon/2}}{|Z|}\). By substitution of this value of \(\gamma\) into (87) we obtain (23). Inequality (24) is justified by (23) and the concavity of the \(\ln\) function. Finally, we prove the lower bound (25):
\[
-\mathbb{E}_{p_Q(Z)} \ln \hat{p}_Q(Z) = \mathbb{E}_{[\hat{p}_Q(Z) - p_Q(Z)]} \ln \hat{p}_Q(Z) - \mathbb{E}_{\hat{p}_Q(Z)} \ln \hat{p}_Q(Z)
\geq -\frac{1}{2} \|\hat{p}_Q(Z) - p_Q(Z)\|_1 \ln \frac{1 + \gamma|Z|}{\gamma} + H(\hat{p}_Q(Z))
\geq H(\hat{p}_Q(Z)) - \sqrt{\varepsilon/2} \ln \frac{|Z|(1 + \sqrt{\varepsilon/2})}{\sqrt{\varepsilon/2}}.
\]

References


