AN ABSTRACT OF THE DISSERTATION OF

Behrouz Behmardi for the degree of Doctor of Philosophy in Electrical and Computer Engineering presented on November 28, 2012.

Title: A probabilistic framework and algorithms for modeling and analyzing multi-instance data

Abstract approved: _____

Raviv Raich

Multi-instance data, in which each object (e.g., a document) is a collection of instances (e.g., word), are widespread in machine learning, signal processing, computer vision, bioinformatic, music, and social sciences. Existing probabilistic models, e.g., latent Dirichlet allocation (LDA), probabilistic latent semantic indexing (pLSI), and discrete component analysis (DCA), have been developed for modeling and analyzing multi-instance data. Such models introduce a generative process for multi-instance data which includes a low dimensional latent structure. While such models offer a great freedom in capturing the natural structure in the data, their inference may present challenges. For example, the sensitivity in choosing the hyper-parameters in such models, requires careful inference (e.g., through cross-validation) which results in large computational complexity. The inference for fully Bayesian models which contain no hyper-parameters often involves slowly converging sampling methods. In this work, we develop approaches for addressing such challenges and further enhancing the utility of such models.

This dissertation demonstrates a unified convex framework for probabilistic modeling of multi-instance data. The three main aspects of the proposed framework are as follows. First, joint regularization is incorporated into multiple density estimation to simultaneously learn the structure of the distribution space and infer each distribution. Second, a novel confidence constraints framework is used to facilitate a tuning-free approach to control the amount of regularization required for the joint multiple density estimation with theoretical guarantees on correct structure recovery. Third, we formulate the problem using a convex framework and propose efficient optimization algorithms to solve it.

This work addresses the unique challenges associated with both discrete and continuous domains. In the discrete domain we propose a confidence-constrained rank minimization (CRM) to recover the exact number of topics in topic models with theoretical guarantees on recovery probability and mean squared error of the estimation. We provide a computationally efficient optimization algorithm for the problem to further the applicability of the proposed framework to large real world datasets. In the continuous domain, we propose to use the maximum entropy (MaxEnt) framework for multi-instance datasets. In this approach, bags of instances are represented as distributions using the principle of MaxEnt. We learn basis functions which span the space of distributions for jointly regularized density estimation. The basis functions are analogous to topics in a topic model.

We validate the efficiency of the proposed framework in the discrete and continuous domains by extensive set of experiments on synthetic datasets as well as on real world image and text datasets and compare the results with state-of-the-art algorithms.

A probabilistic framework and algorithms for modeling and analyzing multi-instance data

by

Behrouz Behmardi

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented November 28, 2012
Commencement June 2013

Doctor of Philosophy dissertation of Behrouz Behmardi presented on November 28, 2012.

APPROVED:

Major Professor, representing Electrical and Computer Engineering

---

Director of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Behrouz Behmardi, Author

Dedicated to my parents

Rostam and Irandokht

ACKNOWLEDGEMENTS

I would like to express my highest gratitude to my adviser professor Raviv Raich who helped me gently along my PhD candidacy. Discussion with him was a source of tremendous insights. I always was inspired by his talent in solving problems and his ability to combine the theory with applications.

I would like to thank all of my committee members for agreeing to be in my committee and for their fruitful comments. I would like to thank Dr. Shiwoo Lee and Dr. Toni Doolen from IE department for their advisory and support during the first two years of my staying at OSU. I want to give a special thank to Dr. Mehran Sepehri for his valuable support and encouragement during my graduate study in Iran.

I would like to thank all of my friends here at OSU who made the university environment friendly and unforgettable for me. In particular, I want to give thank to Sejoon, Juthamas, Weerakit, Kyoung, Madan, Deepthi, Balaji, Gaole, Qi, Greg, Evgenia, Hadi, Mohammad, Javad, and Majid for supporting and helping me.

The last but not the least, I want to give my deepest gratitude and love to my parents Rostam and Iran, my brother Behmard, and my little sister Behnaz. Their unconditional love and support always accompanying me throughout my life. I would like to give a special thank to my wife Arta for always being there, and giving me her infinite love and support.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

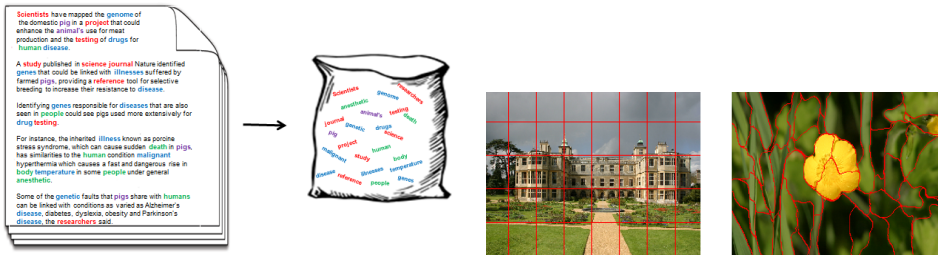
LIST OF FIGURES (Continued)

LIST OF TABLES

LIST OF ALGORITHMS

Chapter 1: Introduction

In multi-instance data each object (bag) is a collection of observations (instances). Multi-instance data appears in a variety of applications in machine learning [23], computer vision [112], bioinformatic [35], music [62], and social sciences [95]. For example, in text document processing a document (bag) can be represented as a collection of words (instances). Bag-of-words representation is a common way of representing text in the corpus of documents [100]. In this representation, first a basic dictionary of unique words (Vocabulary) is constructed by extracting all the words across the documents in the corpus. Then, each document in the corpus is represented as a vector of count of the number of occurrences of each word. The end results is a term-by-document matrix whose rows contain word count for each document in the corpus. Thus, term-by-document representation provides a fix-length vector of integer numbers for an arbitrary length document. In image processing, an image (bag) can be represented as a collection of the local patches or regions (instance) in the image (see Fig. 1.1). Machine learning



(a) (b)

Figure 1.1: Multi-instance representation for (a) image and (b) text documents.

algorithms are described as either *supervised* or *unsupervised*. In the literature, multi-

instance learning (MIL) refers to the prediction problem or supervised learning [3, 36, 42, 112] in which the main goal is to predict the label of an unseen bag, given the label information of the training bags. On the other hand, learning multi-instance data in an unsupervised manner is called grouped data modeling [22, 23, 107] in which the main goal is to uncover the underlying (hidden) structure of the data in the input. In this dissertation, we use multi-instance learning term referring to a class of learning problems where the data is multi-instance. In the following, we review current approaches for learning multi-instance data.

1.1 Different approaches for learning multi-instance data

We review the current approaches for multi-instance learning from different perspectives. First, we categorize multi-instance learning into supervised and unsupervised as well as generative vs. discriminative and discuss approaches in each category. We then sketch some of the existing challenges involving in each approach.

1.1.1 Supervised vs. unsupervised

Multi-instance learning was coined in [42], where drug activity detection was investigated. In their problem, each bag (molecule) is associated with a label and the goal is to predict the label for a previously unseen bag. Formally, supervised multi-instance learning is defined as follows. Suppose we are given a set of N bags $\{(X_1, y_1), (X_2, y_2), \ldots, (X_N, y_N)\}$, where $X \in 2^{\mathcal{X}}$, $\mathcal{X} \subseteq \mathbb{R}^d$ is the feature space, and $y \in \mathcal{Y}$ is either a binary or multi-class label associated with each bag. The instances in bag i are denoted by $x_{i1}, x_{i2}, \ldots, x_{in_i}$, $x_i \in \mathcal{X}$, where n_i is the total number of instance in bag i. The problem

of MIL is to learn a classifier $f : 2^{\mathcal{X}} \rightarrow \mathcal{Y}$. In binary classification ($y_i \in \{-1, +1\}$), a bag is positive if at least one of the instances is positive. Due to the ambiguity of the label information related to instances and the weak association between instance-level information and bag-level information, supervised MIL is a challenging task. Since the introduction of MIL in machine learning and signal processing, numerous algorithms have been proposed either by adapting traditional algorithms to MIL, e.g., Citation-kNN [112], MI-SVM and mi-SVM [3], or by devising a new algorithm specifically for MIL, e.g., axis-parallel rectangles (APR) [42] and diverse density [80, 81].

The main goal in unsupervised learning for multi-instance data is to learn the underlying structure of the data in the input space. Due to the high dimensional nature of objects in multi-instance data (e.g., a usual vocabulary size in a corpus of documents can be about $20,000$), it is beneficial to simplify the representation of objects in multi-instance learning by exploring the inner structure of such datasets. Unsupervised learning algorithms for multi-instance data are based on hidden variable modeling of data. Hidden variable models are structured distributions in which observed data interact with hidden random variables. Latent semantic indexing (LSI) [39], probabilistic latent semantic indexing (pLSI) [63], and latent Dirichlet allocation (LDA) [23] are well-known unsupervised algorithms for leaning multi-instance data.

1.1.2 Generative vs. discriminative

The generative probabilistic approach is commonly used for unsupervised learning of multi-instance. The concept of topic model, a hierarchical Bayesian network, was proposed for uncovering the underlying semantic structure of multi-instance discrete data where each object can be represented as a vector of counts over a fixed size vocabulary

(bag-of-word representation) [23, 63]. Topic models have been applied to many kinds of data such as text documents in text processing [23,57] or images in computer vision [51]. A well-known topic modeling approach is LDA [23,107]. The framework of topic models is extended to supervised learning of multi-instance data by incorporating the label information in the model such as supervised LDA [20,71]. The discriminative multi-instance learning algorithms are the generalization of the traditional margin-based discriminative classifier (e.g., SVM) to the multi-instance case [3, 26, 52, 120]. For example, in kernel methods the calculation of a kernel function between two bags is done by expressing $K(X, X')$ in terms of all the single instance kernel between all the instances from bag X and all the instances from bag X', i.e., $K(X, X')$ for all $(x, x') \in X \times X'$ (e.g., [3, 52]). In extending the k-nearest neighborhood to multi-instance case [112, 120], one needs to develop a generalization of the classical single instance metric (e.g., Euclidean) to a distance between two bags (e.g., Hausdorf distance).

In the probabilistic approach to multi-instance learning, the domain of probabilities can be divided into two basic classes: discrete and continuous. In the discrete domain, the sampling space is the finite or infinite number of countable states. Each object in this domain can be represented as a histogram over a bag-of-word representation. Note that continuous datasets can be discretized using a dictionary approach. In the continuous domain, the sampling space is uncountable. The features in this case can be defined as a real-valued.

1.2 Challenges

Current ongoing efforts toward learning from multi-instance data are focused around *i*) Bayesian inference for fitting a generative model to available data and *ii*) discriminative

learning for multi-instance data in a supervised manner. The first approach allows for a generative probability model which best describes the data but present challenges for computational complexity of Bayesian inference [5]. The second approach does not offer a probabilistic generative model for multi-instance data. Moreover, supervised learning algorithms can be computationally expensive for large datasets. Distance-based MIL algorithms such as Citation-kNN [112] and bag-level kernel SVM [52], construct a bag-level similarity measure that depends on pairwise instance-level similarities.

1.3 Peek at the results

In this dissertation, we provide a convex framework for learning multi-instance datasets in an unsupervised setting, which addresses the aforementioned issues. We investigate the problem of learning multi-instance data in two different domains: discrete and continuous. In the discrete domain where histogram over the bag-of-words can be used to represent each bag, we propose a confidence-constrained rank minimization to estimate the true low-rank term-by-document matrix from the noisy observation. The proposed framework is convex and free of tuning parameters. Moreover, we provide an in-probability bound for the estimation error. In the continuous domain, we propose a maximum entropy based framework for structured leaning of distribution spaces in multi-instance data. We consider the problem of associating each bag with a probability distribution where instances in each bag are generated in an *i.i.d.* fashion from an unknown probability density function. In this framework, each bag is summarized by a fixed-size parameter set, which carries the information about the instances in the bag. We use the maximum entropy framework to construct a concise representation for distributions associated with bag and provide a convex optimization procedure for inference.

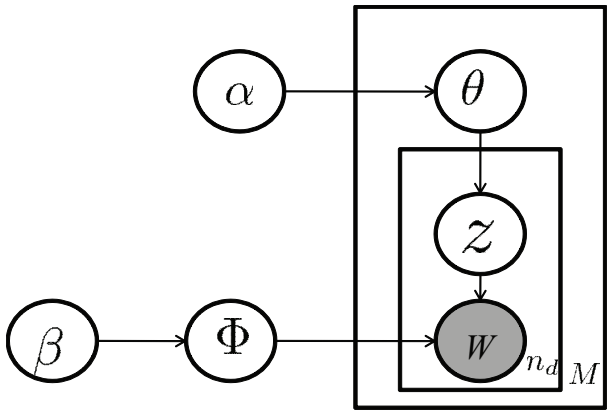
With an m-dimensional parametric representation for each bag, the computational complexity is reduced from $\mathcal{O}(Nn^2)$ to $\mathcal{O}(Nnm)$, where N is the total number of bags, n is the number of instances inside each bag, and m is the dimension of the parameter space. We propose a joint density regularization framework to perform density estimation for multiple densities. Using a sparse representations over a set of basis functions to learn the space of distributions in a non-parametric framework has been studied in [89]. These basis functions provide a continuous analogue to topics in a topic models.

1.4 Background

1.4.1 Probabilistic topic model

Probabilistic topic models are generative models. Topic probabilities provide an explicit representation of documents in probabilistic topic models. The sampling process from this model can be explained as follows.

Each document is drawn in an i.i.d. fashion. For the dth document, $d = \{1, \ldots, M\}$, a random distribution of topics $p(z_{dj} = t|\theta) \triangleq \theta_d(t)$, $t \in \{1, \ldots, T\}$ is drawn. In LDA, $\theta_d \sim \text{Dir}(\alpha)$. Then, for jth word in document d, $j = \{1, \ldots, n_d\}$, a topic assignment z_{dj} is drawn, based on the topic distribution $\theta_d(t)$. Finally, word w_{dj} is drawn based on the conditional distribution $p(w_{dj} = l|z_{dj} = t, \Phi) \triangleq \Phi_{lt}, l = \{1, \ldots, L\}$. Note that Φ is a topics matrix where columns corresponds to topics $\{1, \ldots, T\}$ and rows correspond to vocabulary words. The graphical representation of LDA is shown in Fig. 1.2 and the precise sampling process for LDA is described in Algorithm 1. A key observation in topic models is that the probability distribution of word w_{dj} can be obtained by marginalizing



(a)

Figure 1.2: The graphical model for LDA [105].

the joint word-topic distribution over the topic:

$$p(w_{dj} = l|\theta_d) = \sum_{t=1}^{T} p(w_{dj} = l|z_{dj} = t, \Phi)p(z_{dj} = t|\theta_d). \tag{1.1}$$

To simplify the notation, we represent (1.1) in a matrix format,

$$\Psi = \Phi\theta, \tag{1.2}$$

where $\Psi_{ld} \triangleq p(w_{dj} = l|\theta_d)$, $\Psi \in \mathbb{R}^{L\times M}, \Phi \in \mathbb{R}^{L\times T}$, and $\theta \in \mathbb{R}^{T\times M}$. In other words, the vocabulary term-by-document matrix Ψ can be decomposed into the product of Φ and θ where Φ is the vocabulary probability per topic (topic matrix) and θ is the topic proportion per document. Note that the model in (1.2) is also applicable to pLSI. Columns of these matrices are probability vectors satisfying non-negativity and sum-to-one property. The introduction of latent topic variables allows for reduced dimension representation of the term-by-document matrix Ψ. The rank of the matrix Ψ is the number of topics T. We define the sample term-by-document matrix $\hat{\Psi}$ as follows:

Algorithm 1 Generative process for LDA

for $t = 1$ to T **do**
  Draw $\Phi_t \sim \text{Dirichlet}(\beta)$
end for
for $d = 1$ to M **do**
  Draw $\theta_d \sim \text{Dirichlet}(\alpha)$
  for $j = 1$ to n_d **do**
    Draw $z_{dj} \sim \text{Discrete}(\theta_d)$
    Draw $w_{dj} \sim \text{Discrete}(\phi_{z_{dj}})$
  end for
end for

$$\hat{\Psi}_{ld} = \frac{1}{n_d} \sum_{j=1}^{n_d} I(w_{dj} = l). \tag{1.3}$$

Therefore, $n_d \hat{\Psi}_{\cdot d} \sim \text{multinomial}(\Psi_{\cdot d}, n_d)$ which for notational ease we denote $\hat{\Psi} \sim \text{norm-multinomial}(\Psi, \mathbf{n})$, where $\mathbf{n} = [n_1, \ldots, n_d]$.

LDA is a probabalistic Bayesian framework for modeling multi-instance data in the discrete domain. LDA attempts to summarize multi-instance data and explain the correlation among them by inferring the topic matrix ϕ and θ.

1.4.2 Maximum entropy

The maximum entropy (MaxEnt) framework for density estimation was first proposed by Janes [64] and has been used in many areas of computer science and signal processing including natural language processing [18, 40], species distribution modeling [47, 92], text classification [87, 125], and image processing [102]. The maximum entropy framework [37] finds a unique probability density function (p.d.f) over $\mathcal{X}$ that satisfies the constraints $E_p[\phi(x)] = \alpha$, where $\phi(x) \in \mathbb{R}^m$ is feature transformation defined over $\mathcal{X}$. In

principle, many p.d.f.'s can satisfies the constraints. The maximum entropy approach selects a unique distribution among them. The problem of single density estimation in the maximum entropy framework can be formulated as

$$\begin{aligned} &\text{maximize} \quad H(p) \\ &\text{subject to} \quad E_p[\phi_j] = \alpha_j \\ &\qquad\qquad \int p(x)dx = 1, \end{aligned} \tag{1.4}$$

where $H(p) = -\int p(x) \log p(x) dx$ is the entorpy of $p(x)$ and $E_p[\phi_j] = \int p(x)\phi_j dx$. It can be shown that a solution to (1.4) can be represented as follows:

$$p_\lambda(x) = \exp\big(\lambda^T \phi(x) - Z(\lambda)\big), \tag{1.5}$$

where $Z(\lambda) = \log \int \exp \lambda^T \phi(x)$. There are several algorithms for solving MaxEnt, e.g., iterative scaling [40] and its variants [47,92], gradient descent, Newton, and quasi-Newton approach [78, 99].

In multi-instance data modeling, we can assume that instances inside each bag are *i.i.d.* samples from an unknown density function. Therefore, one can use the principle of maximum entropy approach to fit a distribution to each bag. We use this framework for multi-instance data modeling in the continuous domain.

1.4.3 Nuclear norm

Multi-instance data usually exist in a very high dimensional space. For example, the size of a dictionary in a corpus of text documents can be in the order of 10^4. An efficient way of modeling multi-instance data is to summarize the representation of the data by projecting them into a lower dimensional space. This low dimensional space corresponds to the hidden structure of the data. Rank minimization is an approach in dimension reduction which finds a linear subspace of the observed data by constraining the dimension of the given matrix. In general, rank minimization problems are NP hard [82]. Various algorithms have been proposed to solve the general rank minimization problem locally (e.g., see [58,83]). A heuristic replacement of the rank minimization with a nuclear norm minimization is commonly proposed [50, 97].

The nuclear norm of a matrix is defined as $\|X\|_* = \sum_i \sigma_i$, where $\sigma_i \geq 0$ are the singular values of matrix X given by the following sigular value decomposition $X = U\Sigma V^T$. The nuclear norm is a special class of Schatten norm. The Schatten norm for matrix X is defined as $\|X\|_p = (\sum_i \sigma_i^p)^{\frac{1}{p}}$. When $p = 1, \|X\|_p$ is equal to the nuclear norm, which is the sum of the singular values of matrix X. Similar to the use of l_1-regularization for sparsity, nuclear norm regularization is used to enforce low-rank in the matrix setting and hence can be used to facilitate rank-constrained dimension reduction.

1.5 Dissertation Overview

In **Chapter 2**, we propose a confidence-constrained rank minimization (CRM) to recover the exact number of topics in topic models with theoretical guarantees on recovery probability and mean squared error of the estimation. Topic models have been proposed

to model a collection of data such as text documents and images in which each object (e.g., a document) contains a set of instances (e.g., words). In many topic models, the dimension of the latent topic space (the number of topics) is assumed to be a deterministic unknown. The number of topics significantly affects the prediction performance and interpretability of the estimated topics. We provide a computationally efficient optimization algorithm for the problem to further the applicability of the proposed framework to large real world datasets. Numerical evaluations are used to verify our theoretical results. Additionally, to illustrate the applicability of the proposed framework to practical problems, we provide results in image classification for two real world datasets and text classification for three real world datasets.

In **Chapter 3**, we present a novel entropy estimator for a given set of samples drawn from an unknown probability density function (PDF). Counter to other entropy estimators, the estimator presented here is parametric. The proposed estimator uses the maximum entropy principle to offer an m-term approximation to the underlying distribution and does not rely on local density estimation. The accuracy of the proposed algorithm is analyzed and it is shown that the estimation error is $\mathcal{O}(\log n/n)$. In addition to the analytic results, a numerical evaluation of the estimator on synthetic data as well as on experimental sensor network data is provided. We demonstrate a significant improvement in accuracy relative to other methods.

In **Chapter 4**, we analyze the error of entropy estimation for an unknown density function $p(x)$ using the principle of maximum entropy approach. We propose two estimators for entropy estimation which is called brute-force and greedy m-term approximation. The derivation of the error bound of two estimators is provided.

In **Chapter 5**, we present the maximum entropy (MaxEnt) framework for learning multi-instance data in which each object (bag) is represented as a collection of obser-

vations (instances). In this approach each bag is represented as a distribution using the principle of MaxEnt. We introduce the concept of confidence-constrained MaxEnt (CCMaxEnt) to simultaneously learn the structure of the distribution space and infer each distribution. We learn basis functions which span the space of distributions in CCMaxEnt. The basis functions are analogous to topics in a topic model. We propose KL-divergence for measuring similarities at the bag-level which captures the statistical properties of each bag. In the experimental section, we evaluate the performance of the proposed approach in terms of rank recovery in the space of distributions and compare it with the regularized MaxEnt approach. Moreover, we compare the performance of CCMaxEnt with the state-of-the-art algorithms in multi-instance learning (MIL) and show a comparable results in terms of accuracy with reduced computational complexity.

Chapter 2: On Confidence-Constrained Rank Recovery in Topic Models

2.1 Introduction

In many applications of machine learning, such as text classification, image processing, and web classification, a multi-instance representation of objects is commonly used [4, 118]. In multi-instance datasets, an object is represented as a set of instances or bag of instances instead of a single instance. For example, in a corpus of documents, a document (*object*) comprises of words (*instances*). Often, distributions can be considered to represent multi-instance data. For example, in a multi-instance discrete dataset such as documents, the bag-of-words is a representation of a histogram over a given vocabulary. Due to the high dimensional nature of objects in multi-instance datasets (e.g., a usual vocabulary size in a corpus of documents can be about $20,000$), it is beneficial to simplify the representation of objects in multi-instance datasets by exploring the inner structure of such datasets. The framework of topic models introduces a low dimensional structure by associating documents with a low dimensional distributions over a small set of topics. In the generative approach to topic models, a subset of topics is first selected and the document is generated based on selecting words from the assigned topics. Some of the early well-known topic models are latent semantic indexing (LSI) [39], probabilistic latent semantic indexing (pLSI) [63], and latent Dirichlet allocation (LDA) [23]. We refer the reader to [21] for review on more recent developed topic models.

The number of topics (dimension of the latent space) has a significant effect on the quality of the model and interpretability of the estimated topics [23]. Heuristically, this problem is addressed in the literature by scanning through a range of numbers of topics and comparing performance measures such as perplexity on a held-out dataset or classification accuracy across the range [23, 63, 114]. In [1], it is mentioned that overestimating the number of topics can be remedied by ranking the topics and removing those which are not related to the theme of the data. Bayesian nonparametric topic models [22,53,107] provide a solution using Hierarchical Dirichlet Processes (HDP). The associated Bayesian inference is often regarded as a computationally complex approach [5]. A cross validation approach for selecting the number of topics in topic models is proposed in [66]. While this approach seems to be efficient in number of topics selection, different choices of held-out patterns and sizes have significant impact on the results. Term-by-document matrix is commonly used for data representation in topic models. The number of topics is the rank of such a matrix. Our interest is in devising a provable and computationally efficient method to jointly determine the rank and recover the term-by-document probability matrix from its noisy observation.

Constrained rank recovery of an unknown matrix has been studied vastly in the literature in the communities of signal processing, control system, and machine learning [33,43,79] in problems such as matrix completion [106] and matrix decomposition [28]. While for simple cases singular value decomposition (SVD) has been a common tool, in the constrained setting rank minimization presents additional challenges. One of the main challenges is the non-convex nature of the rank operator. Rank minimization is heuristically replaced with a nuclear norm minimization [30,50,69,97,98]. Nuclear norm minimization can be formulated as a semidefinite programming (SDP) and solved via general SDP solvers such as SDPT3 and SeDuMi. Although the convergence of these

solvers is guaranteed, they can not be applied for a large scale problem due to the high computational complexity of Newton direction [27, 74, 108]. Due to the problem of computational complexity of SDP, several economical approaches have been developed. Most of these approaches are based on the idea of proximal point approximation (Moreau-Yosida regularization [72]) resulting in a closed-form solution for nuclear norm minimization [27, 72, 74, 108]. An Augmented Lagrange multiplier (ALM) [73] is an alternative which proposes to minimize the nuclear norm of the low-rank component plus l_1 norm of the sparse component with augmented Lagrange approach. These methods have been promising in terms of computational complexity. For example, in [73] robust PCA is implemented using only 20 iterations of a highly economical version of SVD. The conditions under which the low-rank matrix with missing entries can be estimated with high probability are proposed in [28,30]. These methods have been applied to video surveillance and image recovery. We are interested in using rank recovery methods to determine the number of topics in topic models. However, we are faced with the following challenges. First, the observed term-by-document matrix is contaminated by a multinomial sampling noise as opposed to Gaussian noise [29, 68] or sparse noise [28]. Our problem includes a specific set of constraints such as positivity and sum-to-one which restrict the search space in the optimization problem.

We present a framework and algorithms for a provable rank recovery in topic models. Specifically, our contributions in this section are as follows: 1) We propose sufficient conditions for exact rank recovery in topic models as a rank minimization problem. 2) We provide a new framework of parameter free confidence-constrained convex optimization as an alternative to rank minimization problem, which can overcome the issues of Bayesian inferences such as *i*) computational complexity associated with sampling methods, *ii*) approximation associated with variational Bayes approach [6], and *iii*)

computational complexity associated with hyperparameter tuning [110]. 3) We provide an analytical evaluation of the sufficient conditions for exact recovery of the number of topics in topic models. Moreover, we provide a bound on the sum of squared errors in terms of the model parameters such as number of documents, vocabulary size, and number of words in each document. 4) We provide an accelerated algorithm to solve the proposed convex optimization problem. We reformulate the problem in the dual form. By evaluating the duality gap, we are able to provide accuracy guarantees for the algorithm. 5) We evaluate our theoretical results on synthetic datasets. 6) Finally, we apply the proposed method on two image datasets and three real world text datasets to illustrate how the method can be applied to perform dimension reduction.

The rest of this section is organized as follows. In Section 2.2, the exact rank recovery in topic models is formulated. Section 2.3 introduces the method of confidence-constrained rank recovery in topic models. Section 2.4 provides the theoretical guarantees for the proposed confidence-constrained rank minimization. In Section 2.5, an accelerated gradient projection method for solving the dual form of confidence-constrained nuclear norm minimization is proposed. In Section 2.6, the evaluation of our theoretical results against the simulation is presented. Section 2.7 illustrates how our method can be applied to image and text datasets.

2.2 Problem formulation

In this section, we present the problem of determining the number of topics in probabilistic topic models. We start with the generative process associated with the probabilistic topic model and then proceed with the formulation of identifying the number of topics in topic models. The theoretical framework for exact rank recovery proposed in this

section can be applied to topic models with the following properties: (i) The generative process involves a multinomial sampling from a probability matrix and (ii) the probability matrix can be decomposed as a product of two probability matrices. We carry out our derivation on the well-known LDA model.

2.2.1 Probabilistic topic models

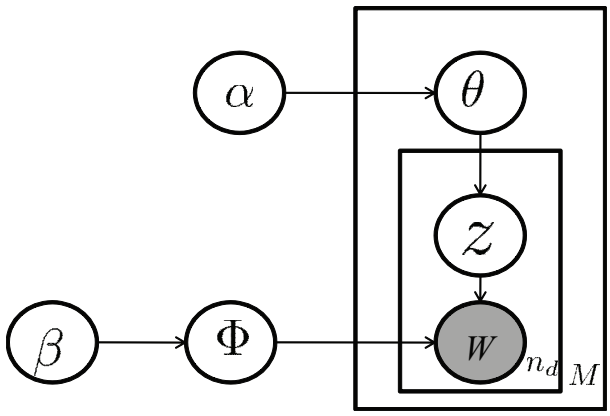
Probabilistic topic models are generative models. Topic probabilities provide an explicit representation of documents in probabilistic topic models. The sampling process from this model can be explained as follows (for a list of notation, we refer the reader to Table 2.1).

Table 2.1: Notation used in this section

Ψ	Term-by-document matrix	θ_d	Per-document topic proportion
$\hat{\Psi}$	Sample term-by-document matrix	Φ	Topics matrix
Ψ_0	Rank minimizing term-by-document matrix	z_{dj}	Per-word per-document topic assignment
M	Number of documents	α	Dirichlet prior parameter for topic proportion
L	Vocabulary size	β	Dirichlet prior for Topics matrix
T	Number of topics (Rank(Ψ))	λ	Lagrangian multiplier
n_d	Number of words in document d	n	$\min(n_d), d = 1, \ldots, M$
σ_T	Smallest non-zero singular value of Ψ		

Each document is drawn in an i.i.d. fashion. For the dth document, $d = \{1, \ldots, M\}$, a random distribution of topics $p(z_{dj} = t|\theta) \triangleq \theta_d(t)$, $t \in \{1, \ldots, T\}$ is drawn. In LDA, $\theta_d \sim \text{Dir}(\alpha)$. Then, for jth word in document d, $j = \{1, \ldots, n_d\}$, a topic assignment z_{dj} is drawn, based on the topic distribution $\theta_d(t)$. Finally, word w_{dj} is drawn based on the conditional distribution $p(w_{dj} = l|z_{dj} = t, \Phi) \triangleq \Phi_{lt}, l = \{1, \ldots, L\}$. Note that Φ is

a topics matrix where columns corresponds to topics $\{1, \ldots, T\}$ and rows correspond to vocabulary words. The graphical representation of LDA is shown in Fig. 2.1 and the precise sampling process for LDA is described in Algorithm 2. A key observation in topic



(a)

Figure 2.1: The graphical model for LDA [105].

models is that the probability distribution of word w_{dj} can be obtained by marginalizing the joint word-topic distribution over the topic:

$$p(w_{dj} = l|\theta_d) = \sum_{t=1}^{T} p(w_{dj} = l|z_{dj} = t, \Phi)p(z_{dj} = t|\theta_d). \tag{2.1}$$

To simplify the notation, we represent (2.1) in a matrix format,

$$\Psi = \Phi\theta, \tag{2.2}$$

where $\Psi_{ld} \triangleq p(w_{dj} = l|\theta_d)$, $\Psi \in \mathbb{R}^{L\times M}, \Phi \in \mathbb{R}^{L\times T}$, and $\theta \in \mathbb{R}^{T\times M}$. In other words, the vocabulary term-by-document matrix Ψ can be decomposed into the product of Φ and θ where Φ is the vocabulary probability per topic (topic matrix) and θ is the topic proportion per document. Note that the model in (2.2) is also applicable to pLSI.

Columns of these matrices are probability vectors satisfying non-negativity and sum-to-one property. The introduction of latent topic variables allows for reduced dimension representation of the term-by-document matrix Ψ. The rank of the matrix Ψ is the number of topics T. We define the sample term-by-document matrix $\hat{\Psi}$ as follows:

Algorithm 2 Generative process for LDA

for $t = 1$ to T **do**
  Draw $\Phi_t \sim$ Dirichlet(β)
end for
for $d = 1$ to M **do**
  Draw $\theta_d \sim$ Dirichlet(α)
  for $j = 1$ to n_d **do**
    Draw $z_{dj} \sim$ Discrete(θ_d)
    Draw $w_{dj} \sim$ Discrete($\phi_{z_{dj}}$)
  end for
end for

$$\hat{\Psi}_{ld} = \frac{1}{n_d} \sum_{j=1}^{n_d} I(w_{dj} = l). \tag{2.3}$$

Therefore, $n_d \hat{\Psi}_{.d} \sim$ multinomial($\Psi_{.d}, n_d$) which for notational ease we denote $\hat{\Psi} \sim$ norm-multinomial($\Psi, \mathbf{n}$), where $\mathbf{n} = [n_1, \ldots, n_d]$.

2.2.2 Topics number recovery

Assume an unknown low-rank term-by-document matrix Ψ is obtained through the process explained in Section 2.2.1. We observe matrix $\hat{\Psi} \sim$ norm-multinomial($\Psi, \mathbf{n}$). Since $\hat{\Psi}$ could be full-rank due to the presence of noise in the sampling process, a straightforward examination of its singular values may not provide an immediate indication on the rank of Ψ. Furthermore, even if rank of the matrix Ψ is available, identifying a

low-rank matrix Ψ which is similar to $\hat{\Psi}$ is a nontrivial problem. Specifically, we are interested in: 1) Estimating the term-by-document matrix Ψ from its noisy observations matrix $\hat{\Psi}$. 2) Quantifying the accuracy of the estimator of Ψ in two aspects: (i) Understanding the conditions under which the exact rank of the true matrix Ψ can be recovered. (ii) Characterizing the estimation error of the matrix Ψ associated with the matrix reconstruction. Note that we propose the estimation of the matrix Ψ rather than the decomposition of Ψ into the product of two probability matrices Φ and θ. While the connection is obvious, the problem of decomposing the estimated low-rank Ψ into the products of two probability matrices presents additional challenges which we reserve for future work.

2.3 Confidence-constrained rank recovery

In this section, we introduce the framework of confidence-constrained rank recovery. We start by describing the maximum likelihood (ML) solution for estimating matrix Ψ from its noisy observation $\hat{\Psi}$. Then, we introduce the regularized ML to address the problem of rank recovery. Finally, we conclude this section with the introduction of confidence-constrained rank minimization approach.

2.3.1 Unconstrained maximum likelihood

The log-likelihood for the probabilistic topic model in (2.1) can be written as follows [63]:

$$\mathcal{L} = \sum_{d=1}^{M} \sum_{l=1}^{L} n_{ld} \log \Psi_{ld}. \tag{2.4}$$

Using the fact that $n_{ld} = n_d \hat{\Psi}_{ld}$, we can rewrite the negative log-likelihood function as follows:

$$\sum_{d=1}^{M} n_d D_{kl}(\hat{\Psi}_{\cdot d} \| \Psi_{\cdot d}) = -\mathcal{L} + \Upsilon, \tag{2.5}$$

where $\Upsilon = \sum_{d=1}^{M} n_d \sum_{l=1}^{L} \hat{\Psi}_{ld} \log \hat{\Psi}_{ld}$ is a constant and $D_{kl}(p\|q) = \sum_k p_k \log \frac{p_k}{q_k}$. Hence, the unconstrained ML estimate of Ψ can be obtained using the following optimization

$$\begin{aligned} \hat{\Psi}_{ML} \quad = \quad & \arg\min_{\tilde{\Psi}} \sum_{d=1}^{M} n_d D_{kl}(\hat{\Psi}_{\cdot d} \| \tilde{\Psi}_{\cdot d}), \\ & \text{subject to} \qquad \tilde{\Psi} \geq 0, \\ & \qquad\qquad\quad 1^T \tilde{\Psi} = 1. \end{aligned} \tag{2.6}$$

Since the ML formulation does not incorporate information on rank of the matrix Ψ, its solution is the trivial $\hat{\Psi}_{ML} = \hat{\Psi}$ solution. In other words, even though the nonnegative $\sum_{d=1}^{M} n_d D_{kl}(\hat{\Psi}_{\cdot d} \| \tilde{\Psi}_{\cdot d})$ can be made zero by setting $\tilde{\Psi} = \hat{\Psi}$, the rank difference $|\text{Rank}(\tilde{\Psi}) - \text{Rank}(\Psi)|$ may be large. The ML approach in its unconstrained formulation advocates the potentially full rank matrix $\hat{\Psi}$ as an estimate for Ψ. In the following, we show how the ML approach can be modified to account for rank constraints using a regularization/penalty term.

2.3.2 Penalized Maximum Likelihood

In this section, we introduce regularized ML, constrained ML, and model order selection (MOS) that potentially can be used to address the problem of rank recovery associated with ML solution. For each framework, we start with the formulation and then

proceed with the corresponding challenges. In contrast to confidence-constrained rank minimization approach which we introduce in the following section, there are no guarantees for exact rank recovery in topic models using penalized ML. Analogous to the use of l_1-regularizer for sparsity, we consider the use of the nuclear norm to enforce the rank constraint in the matrix setting. The heuristic replacement of rank with nuclear norm has been proposed in the literature for matrix completion [29, 97], collaborative filtering [103], and multi-task learning [93].

In regularized ML, a regularized nuclear norm is added to the objective function in (2.6) yielding:

$$\begin{aligned} \text{minimize} \quad & \sum_{d=1}^{M} n_d D_{kl}(\hat{\Psi}_{\cdot d} \| \tilde{\Psi}_{\cdot d}) + \eta \|\tilde{\Psi}\|_*, \\ \text{subject to} \quad & \tilde{\Psi} \geq 0, \\ & 1^T \tilde{\Psi} = 1. \end{aligned} \tag{2.7}$$

The regularization parameter η weighs the nuclear norm. The regularized ML can be viewed as maximum a posteriori (MAP) criterion using a prior distribution over matrix $\tilde{\Psi}$ of the form $Ce^{-\eta\|\tilde{\Psi}\|}$. This is similar to the interpretation of l_1-regularization for sparse recovery as MAP with a Laplacian prior. Since one can apply the Lagrange multipliers framework to replace a constraint with a regularization term, (2.7) can be formulated as constrained ML. The constrained ML formula considers incorporating the nuclear norm as an additional constraint to (2.6):

$$\begin{aligned} \text{minimize} \quad & \sum_{d=1}^{M} n_d D_{kl}(\hat{\Psi}_{\cdot d} \| \tilde{\Psi}_{\cdot d}), \\ \text{subject to} \quad & \|\tilde{\Psi}\|_* \leq \nu, \end{aligned}$$

$$\tilde{\Psi} \geq 0,$$
$$1^T \tilde{\Psi} = 1, \tag{2.8}$$

where $\nu \geq 0$ is a tuning parameter. For each value of η in (2.8) there is a value of ν in (2.7) which produces the same solution [54]. As an alternative to (2.7) and (2.8), MOS can be applied to rank estimation of a matrix [91, 113]. MOS offers a way to evaluate the classical trade-off between goodness of fit and model complexity. For $r = 1, 2, \ldots, \min(L, M)$, a sequence of optimization problems in the form of (2.6) subject to rank $= r$ is solved to obtain $\tilde{\Psi}^{*(r)}$. Then for each rank r, a cost function including negative log-likelihood at $\tilde{\Psi}^{*(r)}$ plus a penalty term pen(r) is evaluated. The penalty term corresponds to the complexity of the model and is measured based on an information criterion such as Akaike Information Criterion (AIC) or Minimal Description Length (MDL) [91, 113]. Note that in AIC the penalty term corresponds to the number of free parameters in the model. In MDL, each model candidate is assigned with a code length and minimum code length is used for model selection. In some implementations of MDL, each model is assigned with a prior probability and the model that yields the maximum posterior probability is selected. The use of rank minimization for model order selection in system identification is proposed in [75, 84]. Furthermore in [75], the authors proposed the heuristic replacement of the rank with the nuclear norm and showed that it makes the selection of an appropriate model order easier. In the following discussion, we illustrate some of the challenges associated with regularized ML, constrained ML, and MOS proposed in this section.

Discussion One of the challenges associated with the regularized and constrained ML is the choice of the regularization parameters (η and ν, respectively). Often, a criterion for selecting a value for the regularization parameters that guarantees exact

rank recovery of matrix Ψ is unavailable. For the problem of low-rank matrix estimation in the noisy setting, asymptotic relationship between the regularization parameter and estimation accuracy is proposed in [7, 85]. Such results cannot be applied directly to our problem for the following reason. Counter to the sampling process in Section 2.2.1, the sampling process proposed in [85] follows an *i.i.d.* model without the positivity and sum-to-one. In MOS approach, solving the sequence of an optimization problem with rank constraint and evaluating the cost function for different value of rank ($r = 1, 2, \ldots, \min(L, M)$) is computationally complex. While in the unconstrained setting SVD provides a one-shot solution [113], in the constrained setting rank minimization is NP-hard [83]. The heuristic replacement of rank with nuclear norm in MOS proposed in [75,84] suggests a regularization parameter framework. However, no recipe is provided for selecting the regularization parameter to guarantee rank recovery. In the following, we define the confidence-constrained rank minimization and show how our formulation of the problem can address the issues associated with parameter tuning in regularized ML and constrained ML and exhaustive rank search for MOS stated in this section.

2.3.3 Confidence-constrained rank minimization

We consider the concept of the confidence-constrained rank minimization for rank recovery in topic models. Using the statistical formulation of the problem proposed in Section 2.2, an in-probability bound on the objective function in (2.6) can be obtained. The probability bound on data fit criterion allows us to define a confidence set. Confidence set is a high-dimensional generalization of the confidence interval and restricts the search space of the problem. Search inside the confidence set guarantees a low-rank solution. Hence, in this approach the roles of ML objective and rank constrained

are replaced. We consider rank minimization subject to ML objective constraint. The confidence-constrained rank minimization is given by:

$$\begin{aligned} \text{minimize} \quad & \text{Rank}(\tilde{\Psi}) \\ \text{subject to} \quad & \sum_{d=1}^{M} n_d D_{KL}(\hat{\Psi}_{d\cdot} \| \tilde{\Psi}_{d\cdot}) \leq \epsilon(\delta), \\ & \tilde{\Psi} \geq 0, \\ & 1^T \tilde{\Psi} = 1, \end{aligned} \tag{2.9}$$

where $\epsilon(\delta)$ is an in-probability bound for the estimation error. Note in this formulation the tuning parameter $\epsilon(\delta)$ can be obtained by bounding $\sum_{d=1}^{M} n_d D_{KL}(\hat{\Psi}_{d\cdot} \| \tilde{\Psi}_{d\cdot})$. Intuitively the KL confidence-constrained set in (2.9) includes the matrix Ψ, and hence it is guaranteed (w.p. $1-\delta$) that the rank of the solution to (2.9) is less than or equal to the rank of matrix Ψ. The main problem with KL divergence between two matrices is that there is no straightforward way of translating it to the distance between their singular values. Since singular values are related to the rank of a matrix, it is hard to provide the theoretical guarantees for rank recovery in the KL version of the confidence-constrained set. While the KL confidence-constrained formulation is difficult to handle, the Frobenius-norm confidence-constrained formulation provides a convenient framework for proving rank recovery in topic models. The problem of parameter tuning is elegantly addressed in this framework by obtaining a model based in-probability uniform bound on the confidence set. Moreover, the approach does not require a scan through a range of rank values. In the following, we show that in the Frobenius-norm confidence-constrained rank minimization exact rank recovery can be guaranteed.

2.4 Exact rank recovery: theoretical guarantees

In this part, we introduce Frobenius-norm confidence-constrained rank recovery and provide the theoretical guarantees for exact rank recovery in topic models. The KL-divergence confidence-constrained rank recovery in (2.9) is replaced with Frobenius norm confidence-constrained rank recovery since the theoretical results can be shown for the Frobenius-norm case while such results are unavailable for the KL-divergence.

2.4.1 Frobenius-norm confidence-constrained rank minimization (CRM)

For the problem defined in Section 2.2.2, we propose the following confidence-constrained rank minimization:

$$\begin{aligned} \text{(CRM):} \quad & \text{minimize} \quad \text{Rank}(\tilde{\Psi}) \\ & \text{subject to} \quad \|\tilde{\Psi} - \hat{\Psi}\|_F \leq \epsilon(\delta_k), \\ & \qquad\qquad \tilde{\Psi} \geq 0, \\ & \qquad\qquad 1^T \tilde{\Psi} = 1. \end{aligned} \tag{2.10}$$

where

$$\epsilon(\delta_k) = \epsilon^*(\delta_k) \triangleq \sqrt{\frac{1}{n}\left(M + k\sqrt{\frac{M}{2}(1+\frac{3}{n})}\right)}, \quad \delta_k = \frac{1}{1+k^2}, \tag{2.11}$$

where $n_d = n$ for all d. In Appendix 2.8.2, ϵ^* is developed for the general case where document d has n_d words. Here for simplicity, we present the case where $n_d = n$. The parameter $k = \sqrt{\delta_k^{-1} - 1}$ is the number of standard deviation away from the mean, e.g., for $k = 3$, with probability $1 - 1/(1 + k^2) = 0.9$, $\|\tilde{\Psi} - \hat{\Psi}\|_F \leq \epsilon(\delta_3)$ where

$\epsilon(\delta_3) = \sqrt{\frac{1}{n}(M + 3\sqrt{\frac{M}{2}(1+\frac{3}{n})})}$. Note that (2.10) is free of tuning parameters for the following reason. Since the samples are governed by a multinomial distribution, an in-probability bound on the estimation error of the form $\|\Psi - \hat{\Psi}\|_F \leq \epsilon(\delta_k)$ w.p. $1-\delta$ can be obtained. Moreover, since the true low-rank matrix Ψ satisfies the Frobenius norm inequality constraint w.p. $1-\delta$, then Ψ_0 the solution to (2.10) is of equal or lower rank to that of Ψ. While this result is straightforward, the following theorem shows that in fact the CRM solution Ψ_0 has the same rank as Ψ. Moreover, theorem provides a bound on the estimation error [13].

Theorem 1 *Let Ψ be a γ-distinct rank T matrix and $\hat{\Psi} \sim$ norm-multinomial$(\Psi, \boldsymbol{n})$. Assume $\gamma > 2\epsilon$, and $\epsilon = \epsilon^*$ defined in (2.11). Then, with probability at least $1-\delta_k$, Ψ_0 the solution to (2.10) satisfies:*

1. *$\Psi_0 \in 2\epsilon$-neighborhood of Ψ,*
2. *Rank$(\Psi_0) = T$.*

Theorem 1 characterizes Ψ_0 the solution to CRM in (2.10). First, Ψ_0 is at most 2ϵ away from the true matrix Ψ. Theorem 1 is formulated with specific ϵ in (2.11) which comes from the statistical model presented in Section 2.2. With ϵ in (2.11), the Frobenius norm of the estimation error $(\Psi_0 - \Psi)$ is $\mathcal{O}(\sqrt{M/n})$. The second property asserts that under the hypothesis of the Theorem 1, it is guaranteed that with probability $1-\delta$ Ψ_0 has the same rank as the rank of the true unknown matrix Ψ. In other words, the exact rank of the true matrix Ψ can be recovered by solving the CRM optimization problem in (2.10). We now proceed with the proof of Theorem 1. For this, first we provide a detail framework as follows:

Definition 2 *Ψ' is a γ-distinct rank r matrix if $\sigma_1(\Psi') \geq \sigma_2(\Psi') \geq \ldots \geq \sigma_r(\Psi') > \gamma > \sigma_{r+1}(\Psi') = \ldots = \sigma_L(\Psi') = 0$, where σ_i is the i^{th} largest singular value of matrix Ψ'.*

In other words, Ψ' is γ-distinct if all of its non zero singular values are greater than γ.

Definition 3 *Matrix Ψ' is in the ζ-neighborhood of matrix Ψ if $\|\Psi - \Psi'\|_F \leq \zeta$.*

Lemma 4 *W.p. $1-\delta$ matrix Ψ satisfies $\|\Psi - \hat{\Psi}\|_F \leq \epsilon$, where $\epsilon = \epsilon^*$ is given by (2.11).*

Proof See Appendix 2.8.2. ■

Lemma 4 guarantees that w.p. $1-\delta$ the confidence-constrained set $S(\hat{\Psi}, \epsilon^*) = \{\Psi' \mid \|\hat{\Psi} - \Psi'\|_F \leq \epsilon\}$ contains the true low-rank matrix Ψ.

Lemma 5 *Let Ω be γ-distinct rank r matrix. Then there exists no matrix in the γ-neighborhood of Ω, with the rank $r_0 < r$.*

Proof Suppose $\exists \Omega'$ in the γ-neighborhood with rank $r_0 < r$, therefore

$$\begin{aligned} \gamma &\geq \|\Omega' - \Omega\|_F \\ &\geq \min_{\text{Rank}(\tilde{\Omega})=r_0} \|\tilde{\Omega} - \Omega\|_F. \end{aligned} \tag{2.12}$$

By Eckart-Young theorem [104] the closest $\tilde{\Omega}$ with rank r_0 to Ω in the Frobenius norm is $\tilde{\Omega} = U\Sigma^* V^T$, where $\Omega = U\Sigma V^T$ and $\Sigma^* = \text{diag}(\sigma_1, \ldots, \sigma_{r_0}, 0, \ldots, 0)$. For such $\tilde{\Omega}$, $\|\tilde{\Omega} - \Omega\|_F^2 = \sum_{i=r_0+1}^{r} \sigma_i^2$. Thus, $\gamma \geq \sqrt{\sum_{i=r_0+1}^{r} \sigma_i^2} \geq \sigma_r(\Omega)$. By contradiction to the assumption that $\sigma_r(\Omega) > \gamma$, there exists no such Ω' in γ-neighborhood with rank lower than r. ■

Based on Lemma 5, the γ-distinct property of matrix Ψ assures that all the matrices inside the γ-neighborhood of matrix Ψ have a rank greater than or equal to rank of

matrix Ψ . Using Definitions 2 and 3 and Lemmas 4 and 5, we proceed with the proof of Theorem 1.

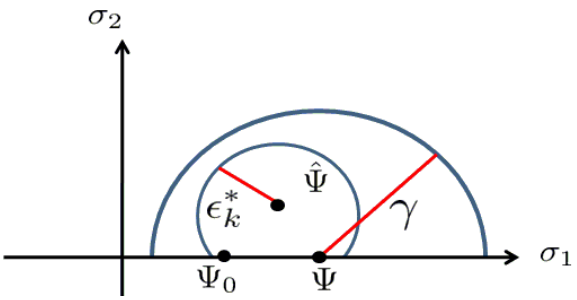
Proof 1) Using the triangle inequality, we have

$$\|\Psi_0 - \Psi\|_F \quad \leq \quad \|\Psi_0 - \hat{\Psi}\|_F + \|\hat{\Psi} - \Psi\|_F. \tag{2.13}$$

Note that the first term on the RHS of (2.13) is less than ϵ with probability 1, since Ψ_0 the solution to (2.10) satisfies the confidence-constraint. Thus, $\Psi_0 \in \epsilon$-neighborhood of $\hat{\Psi}$. The second term on the RHS of (2.13) is a random quantity which can be bounded by ϵ with probability $1-\delta$ by Lemma 4. Therefore $\|\Psi_0 - \Psi\|_F \leq 2\epsilon$ with probability $1-\delta$. ■

Proof 2) Since Ψ_0 is in the 2ϵ-neighborhood of Ψ and $2\epsilon < \gamma$, then Ψ_0 is also in the γ-neighborhood of Ψ. Hence, based on Lemma 5 $\text{Rank}(\Psi_0) \geq \text{Rank}(\Psi)$. On the other hand, since $\Psi \in \epsilon$-neighborhood of $\hat{\Psi}$ w.p. $1 - \delta_k$, and Ψ_0 is the minimum rank solution matrix in ϵ-neighborhood of $\hat{\Psi}$, then $\text{Rank}(\Psi_0) \leq \text{Rank}(\Psi)$. The inequalities can hold only if $\text{Rank}(\Psi_0) = \text{Rank}(\Psi) = T$. ■

Discussion The basic idea of Theorem 1 relies on two main principles. 1) γ-distinct property of matrix Ψ which corresponds to the robustness of Ψ to the sampling noise. If γ is large, the matrix Ψ is robust enough to be rank recoverable given a small sampling noise (for illustration see Fig. 2.2). 2) The second principle associates with the magnitude of the sampling noise which controls the size of the confidence-constrained set. Since the statistics of the sampling noise is known, it provides the theoretical guarantees for recovering the exact rank of the matrix Ψ.



(a)

Figure 2.2: This figure shows two sets: *i*) ϵ-neighborhood of matrix $\hat{\Psi}$ (confidence-constrained set) which is defined as $\{\Psi | \|\hat{\Psi} - \Psi\|_F \leq \epsilon\}$ and *ii*) γ-neighborhood of matrix Ψ which is defined as $\{\Psi' | \|\Psi - \Psi'\|_F \leq \gamma\}$. In this figure, matrix Ψ is γ distinct and $\gamma > 2\epsilon_k^*$. Thus, the assumptions of Theorem 1 hold. As a result, Ψ_0 will have the same rank as matrix Ψ.

2.4.2 Confidence-constrained nuclear norm minimization (CNM)

In general, rank minimization problems are NP hard [82]. Various algorithms have been proposed to solve the general rank minimization problem locally (e.g., see [58, 83]). A heuristic replacement of the rank minimization with a nuclear norm minimization is commonly proposed [50, 97]. The nuclear norm of a matrix is defined as $\|X\|_* = \sum_i \sigma_i$ where $\sigma_i \geq 0$ are the singular values of matrix X. The nuclear norm is a special class of Schatten norm. The Schatten norm for matrix X is defined as $\|X\|_p = (\sum_i \sigma_i^p)^{\frac{1}{p}}$. When $p = 1, \|X\|_p$ is equal to the nuclear norm, which is the sum of the singular values of matrix X. Similar to the use of l_1-regularization for sparsity, nuclear norm regularization is used to enforce low-rank in the matrix setting. To solve the rank minimization problem proposed in (2.10), we propose the widely used approach of replacing the rank minimization with the tractable convex optimization problem of nuclear norm minimization. In Section 2.6, we provide the evaluation of CNM only, due to the prohibitive computation complexity associated with CRM. In the following, confidence-constrained nuclear norm

minimization (CNM) is proposed as a convex alternative to (2.10):

$$\begin{aligned} \text{(CNM):} \quad & \text{minimize} && \|\tilde{\Psi}\|_* \\ & \text{subject to} && \|\tilde{\Psi} - \hat{\Psi}\|_F \leq \epsilon, \\ & && \tilde{\Psi} \geq 0, \\ & && 1^T \tilde{\Psi} = 1. \end{aligned} \tag{2.14}$$

We denote the solution to (2.14) by $\tilde{\Psi}^*$. Since the nuclear norm is a convex function, and the set of the inequality and equality constraints construct a convex set, (2.14) is a convex optimization problem. This formulation targets the problem of exact rank recovery for probability matrices under the sampling process described in Section 2.2.1.

2.5 Confidence-constrained nuclear norm minimization algorithm (CNMA)

The nuclear norm minimization problem can be reformulated as an SDP [50]. Off-the-shelf SDP solvers such as SDPT3 and SeDuMi are used to solve this problem. Such software packages use the interior point method with Newton direction which is computationally expensive [27,74,108]. The SDP problem of CNM has $(M+L)\times(M+L)$ semidefinite constraints and $(ML + M + 1)$ equality and inequality constraints. The computational complexity is $\mathcal{O}(\min\{M, L\})^6$ and the memory requirement is $\mathcal{O}(\min\{M, L\})^4$. So while the reformulation is theoretically appealing, computational challenges remain. In the following, we provide an accelerated projection gradient algorithm to solve the dual formulation of CNM. We start with the dual formulation of CNM and then solve it with the

gradient projection approach [19]. We propose an accelerated version of our algorithm using two point approximation [86] and a highly economical SVD-based implementation.

2.5.1 Dual formulation background

We solve (2.14) through formulating the dual problem. Generally, the dual formulation of a problem in the form of

$$\begin{aligned} &\text{minimize} && f_0(x) \\ &\text{Subject to} && f_1(x) \leq 0 \\ & && h(x) = 0, \end{aligned}$$

can be obtained first by constructing the Lagrangian $\mathcal{L}(x, \lambda_1, \underline{\lambda}_2)$ as follows:

$$\mathcal{L}(x, \lambda_1, \lambda_2) = f_0(x) + \lambda_1^T f_1(x) + \lambda_2^T h(x),$$

where $\lambda_1 \geq 0$ and λ_2 are the Lagrange multipliers for the set of inequality and equality constraints, respectively. The Lagrangian incorporates the constraints into the objective function using the Lagrange multipliers λ_1, and λ_2. The second step is to minimize the Lagrangian $\mathcal{L}(x, \lambda_1, \lambda_2)$ with respect to the primal objective variable x. Define $x^*(\lambda_1, \lambda_2)$ as:

$$x^*(\lambda_1, \lambda_2) = \arg\min_x \mathcal{L}(x, \lambda_1, \lambda_2).$$

By replacing $x^*(\lambda_1, \lambda_2)$ in the Lagrangian, we obtain the dual:

$$g(\lambda_1, \lambda_2) = \mathcal{L}(x^*(\lambda_1, \lambda_2), \lambda_1, \lambda_2).$$

The dual formulation is given by the following optimization

$$\begin{aligned} &\text{maximize} \quad && g(\lambda_1, \lambda_2) \\ &\text{Subject to} \quad && \lambda_1 \geq 0. \end{aligned}$$

The dual formulation of the optimization problem has several advantages. First, it provides a lower bound for the primal problem. One can show for any feasible point $\tilde{x}$ in the primal problem, $g(\lambda_1, \lambda_2) \leq f(\tilde{x})$. If the primal problem is convex and the set of inequalities is strictly satisfied for some point inside the feasibility set, then based on Slater's condition the strong duality holds [25]. Hence, the duality gap $f(\tilde{x}) - g(\lambda_1, \lambda_2)$ provides means of assessing convergence of the optimization algorithm. Furthermore, the positivity constraint in the dual formulation can be handled using a simple projection onto the positive orthant. Note that in the primal formulation the projection onto the set of equality and inequality constraints could be more complex.

2.5.2 Dual formulation of CNM

We follow the steps explained in Section 2.5.1. First, we construct the Lagrangian of (2.14) to obtain the dual formulation [12]. The Lagrangian $\mathcal{L}(\tilde{\Psi}, \lambda_1, \underline{\lambda}_2, \Lambda_3)$ for problem in (2.14) can be written as

$$\mathcal{L}(\tilde{\Psi}, \lambda_1, \underline{\lambda}_2, \Lambda_3) = \|\tilde{\Psi}\|_* + \frac{\lambda_1}{2}(\|\tilde{\Psi} - \hat{\Psi}\|_F^2 - \epsilon^2) + \underline{\lambda}_2^T(1 - \tilde{\Psi}^T 1) - \mathrm{tr}(\Lambda_3^T \tilde{\Psi}), \qquad (2.15)$$

where $\lambda_1 \in \mathbb{R}^+$, $\underline{\lambda}_2 \in \mathbb{R}^{M\times 1}$, and $\Lambda_3 \in \mathbb{R}^{+L\times M}$. If we minimize $\mathcal{L}(\tilde{\Psi}, \lambda_1, \underline{\lambda}_2, \Lambda_3)$ with respect to $\tilde{\Psi}$, we obtain $\tilde{\Psi}^*(\lambda_1, \underline{\lambda}_2, \Lambda_3)$. We start by rewriting (2.15) as follows:

$$\mathcal{L}(\tilde{\Psi}, \lambda_1, \underline{\lambda}_2, \Lambda_3) \quad = \quad \|\tilde{\Psi}\|_* + \frac{\lambda_1}{2}\|\tilde{\Psi} - \Psi'\|_F^2 + C(\lambda_1, \underline{\lambda}_2, \Lambda_3), \tag{2.16}$$

where $\Psi' = \hat{\Psi} + \frac{1\underline{\lambda}_2^T}{\lambda_1} + \frac{\Lambda_3}{\lambda_1}$, and $C(\lambda_1, \underline{\lambda}_2, \Lambda_3) = -\frac{\lambda_1}{2}\|\Psi'\|_F^2 + \frac{\lambda_1}{2}\|\hat{\Psi}\|_F^2 + \underline{\lambda}_2^T 1 - \frac{\lambda_1}{2}\epsilon^2$. The solution to the minimization of (2.16) w.r.t. $\tilde{\Psi}$ is

$$\tilde{\Psi}^*(\lambda_1, \underline{\lambda}_2, \Lambda_3) = D_{\frac{1}{\lambda_1}}(\Psi'),$$

where $D_\tau(X)$ is the soft thresholding operator on the singular value of matrix X (for proof see [27]) defined by $D_\tau(X) = U(S - \tau I)_+ V^T$, where $X = USV^T$ is the SVD of X. To obtain the dual, we substitute $\tilde{\Psi}^*(\lambda_1, \underline{\lambda}_2, \Lambda_3)$ back into (2.16), simplify and obtain

$$f(\lambda_1, \underline{\lambda}_2, \Lambda_3) = -\frac{\lambda_1}{2}\|D_{\frac{1}{\lambda_1}}(\Psi')\|_F^2 + \frac{\lambda_1}{2}\|\hat{\Psi}\|_F^2 + \underline{\lambda}_2^T 1 - \frac{\lambda_1}{2}\epsilon^2.$$

Thus the dual formulation of the CNM problem in (2.14) is

$$\begin{aligned} \text{maximize} \quad & f(\lambda_1, \underline{\lambda}_2, \Lambda_3) \\ \text{subject to} \quad & \lambda_1 \geq 0 \\ & \Lambda_3 \geq 0, \end{aligned}$$

where $\lambda_1 \in \mathbb{R}$, $\underline{\lambda}_2 \in \mathbb{R}^{M\times 1}$, and $\Lambda_3 \in \mathbb{R}^{L\times M}$. The positivity for matrix Λ_3 is elementwise. Rather than maximize the dual function, we proceed with the convex minimization of the negative dual, $\tilde{f}(\lambda_1, \underline{\lambda}_2, \Lambda_3) = -f(\lambda_1, \underline{\lambda}_2, \Lambda_3)$.

2.5.3 Gradient projection algorithm for CNM

The CNM optimization problem is expressed as follows:

$$\begin{aligned} \text{minimize} \quad & \tilde{f}(\lambda_1, \underline{\lambda}_2, \Lambda_3) \\ \text{subject to} \quad & \lambda_1 \geq 0 \\ & \Lambda_3 \geq 0, \end{aligned} \tag{2.17}$$

where $\tilde{f}(\lambda_1, \underline{\lambda}_2, \Lambda_3) = \frac{\lambda_1}{2}\|D_{\frac{1}{\lambda_1}}(\hat{\Psi} + \frac{1\underline{\lambda}_2^T}{\lambda_1} + \frac{\Lambda_3}{\lambda_1})\|_F^2 - \frac{\lambda_1}{2}\|\hat{\Psi}\|_F^2 - \underline{\lambda}_2^T 1 + \frac{\lambda_1}{2}\epsilon^2$. We consider the gradient projection method to solve (2.17). The gradient projection method for minimizing a continuous convex function over a closed convex set was proposed in [55]. The modified backtracking approach for the gradient projection method was defined in [19]. Application of the gradient projection method to our problem consists of the following iterations:

$$\begin{aligned} &\lambda_1^{k+1} = [\lambda_1^k - t^k \nabla \tilde{f}_{\lambda_1^k}(\lambda_1, \underline{\lambda}_2, \Lambda_3)]_+, \quad \underline{\lambda}_2^{k+1} = \underline{\lambda}_2^k - t^k \nabla \tilde{f}_{\underline{\lambda}_2^k}(\lambda_1, \underline{\lambda}_2, \Lambda_3) \\ &\Lambda_3^{k+1} = [\Lambda_3^k - t^k \nabla \tilde{f}_{\Lambda_3^k}(\lambda_1, \underline{\lambda}_2, \Lambda_3)]_+, \end{aligned}$$

where $[x]_+ = x$ for $x \geq 0$, and otherwise is zero, $\nabla \tilde{f}_{\lambda_i}(\lambda_1, \lambda_2, \Lambda_3)$ is the gradient with respect to λ_1, $\underline{\lambda}_2$, Λ_3, and t^k is the step size. Note that since the positivity of λ_1 and Λ_3 can be enforced coordinatewise, the projection is trivial. The gradient of $\tilde{f}(\underline{\lambda})$ with respect to $\lambda_1, \underline{\lambda}_2$, and Λ_3 is respectively,

$$\begin{aligned} \nabla \tilde{f}_{\lambda_1}(\lambda_1, \underline{\lambda}_2, \Lambda_3) &= \frac{1}{2}\|D_{\frac{1}{\lambda_1}}(\Psi')\|_F^2 + \frac{1}{\lambda_1}\|D_{\frac{1}{\lambda_1}}(\Psi')\|_* - \frac{1}{\lambda_1}\mathrm{tr}((1\underline{\lambda}_2^T + \Lambda_3)^T D_{\frac{1}{\lambda_1}}(\Psi')) \\ &- \frac{1}{2}\|\hat{\Psi}\|_F^2 + \frac{\epsilon^2}{2}, \end{aligned}$$

$$\begin{aligned}\nabla \tilde{f}_{\underline{\lambda}_2}(\lambda_1, \underline{\lambda}_2, \Lambda_3) &= D_{\frac{1}{\lambda_1}}(\Psi')^T 1 - 1,\\ \nabla \tilde{f}_{\Lambda_3}(\lambda_1, \underline{\lambda}_2, \Lambda_3) &= D_{\frac{1}{\lambda_1}}(\Psi').\end{aligned}$$

The derivative of $\tilde{f}$ with respect to λ_1 is given by $\frac{d}{d\lambda_1}(\frac{\lambda_1}{2}\|D_{\frac{1}{\lambda_1}}(\Psi')\|_F^2) - \frac{1}{2}\|\hat{\Psi}\|_F^2 + \frac{\epsilon^2}{2}$. The derivation of the term $\frac{d}{d\lambda_1}(\frac{\lambda_1}{2}\|D_{\frac{1}{\lambda_1}}(\Psi')\|_F^2)$ which leads to the explicit expression of $\nabla \tilde{f}_{\lambda_1}(\lambda_1, \underline{\lambda}_2, \Lambda_3)$ is provided in Appendix 2.8.1. Upon convergence of the Lagrange multipliers $[\lambda_1, \underline{\lambda}_2, \Lambda_3]$, one can compute the primal objective parameters using $\tilde{\Psi} = D_{\frac{1}{\lambda_1}}(\hat{\Psi} + \frac{1\underline{\lambda}_2^T}{\lambda_1} + \frac{\Lambda_3}{\lambda_1})$. In the following, we first show how to choose the step size for the gradient method using the backtracking approach. Then, we provide the accelerated gradient projection method.

2.5.3.1 Step size

To choose the step size t^k, we use the backtracking approach for gradient projection [19]. The backtracking line search for gradient projection requires the smallest nonnegative integer m_k such that

$$\tilde{f}\bigg(\lambda_1^k(t^k), \underline{\lambda}_2^k(t^k), \Lambda_3^k(t^k)\bigg) \leq \tilde{f}(\lambda_1^k, \underline{\lambda}_2^k, \Lambda_3^k) - \gamma\bigg(\nabla \tilde{f}_{\lambda_1}\Delta\lambda_1^k + \nabla \tilde{f}_{\underline{\lambda}_2}^T \Delta\underline{\lambda}_2^k + \mathrm{tr}(\nabla \tilde{f}_{\Lambda_3}^T \Delta\Lambda_3^k)\bigg),$$

where $\Delta\lambda_1^k = \lambda_1^k - \lambda_1^k(t^k)$, $\Delta\underline{\lambda}_2^k = \underline{\lambda}_2^k - \underline{\lambda}_2^k(t^k)$, $\Delta\Lambda_3^k = \Lambda_3^k - \Lambda_3^k(t^k)$, $t^k = \eta^{m_k} t^0$, $\gamma \in (0, 0.5)$, $t^0 > 0$, and $\eta \in (0, 1)$. The proposed backtracking approach in (2.18) finds a step size t^k which reduces the objective function sufficiently. However to avoid making a small step in each iteration, we start with a large enough step size t^0 which satisfies the following

condition:

$$\tilde{f}\bigg(\lambda_1^k(t^0), \underline{\lambda}_2^k(t^0), \Lambda_3^k(t^0)\bigg) > \tilde{f}(\lambda_1^k, \underline{\lambda}_2^k, \Lambda_3^k) - \gamma\bigg(\nabla\tilde{f}_{\lambda_1}\Delta\lambda_1^k + \nabla\tilde{f}_{\underline{\lambda}_2}^T\Delta\underline{\lambda}_2^k + \mathrm{tr}(\nabla\tilde{f}_{\Lambda_3}^T\Delta\Lambda_3^k)\bigg).$$

Algorithm 3 Accelerated CNMA for exact rank recovery

Choose $\lambda_1^0 = \lambda_1^1 > 0, \underline{\lambda}_2^0 = \underline{\lambda}_2^1 = 0, \Lambda_3^0 = \Lambda_3^1 = 0$, $a_0 = a_1 = 1$, $\eta \in (0,1)$, $\gamma \in (0, 0.5)$, $\mu > 1$, $t_0 > 0$, K, υ

for $k = 1$ to K **do**

$\bar{\lambda}_1^k = \lambda_1^k + \frac{a_{k-1}-1}{a_k}(\lambda_1^k - \lambda_1^{k-1})$, $\bar{\underline{\lambda}}_2^k = \underline{\lambda}_2^k + \frac{a_{k-1}-1}{a_k}(\underline{\lambda}_2^k - \underline{\lambda}_2^{k-1})$, $\bar{\Lambda}_3^k = \Lambda_3^k + \frac{a_{k-1}-1}{a_k}(\Lambda_3^k - \Lambda_3^{k-1})$ {Acceleration}

$\Psi'^k = \hat{\Psi} + \frac{1\bar{\underline{\lambda}}_2^{kT}}{\bar{\lambda}_1^k} + \frac{\bar{\Lambda}_3^k}{\bar{\lambda}_1^k}$

$(U, S, V^T) = \mathrm{svd}(\Psi'^k)$

$\tilde{\Psi}^{k+1} = U(S - 1/\bar{\lambda}_1^k)_+ V^T$ {Soft thresholding}

while $\tilde{f}\bigg(\lambda_1^k(t^0), \underline{\lambda}_2^k(t^0), \Lambda_3^k(t^0)\bigg) \leq \tilde{f}(\bar{\lambda}_1^k, \bar{\underline{\lambda}}_2^k, \bar{\Lambda}_3^k) - \gamma\bigg(\nabla\tilde{f}_{\bar{\lambda}_1}\Delta\bar{\lambda}_1^k + \nabla\tilde{f}_{\bar{\underline{\lambda}}_2}^T\Delta\bar{\underline{\lambda}}_2^k + \mathrm{tr}(\nabla\tilde{f}_{\bar{\Lambda}_3}^T\Delta\bar{\Lambda}_3^k)\bigg)$ **do**

$t^0 = \mu^{n_k} t_0$ {line search (wolf condition)}

end while

while $\tilde{f}\bigg(\lambda_1^k(t^k), \underline{\lambda}_2^k(t^k), \Lambda_3^k(t^k)\bigg) > \tilde{f}(\bar{\lambda}_1^k, \bar{\underline{\lambda}}_2^k, \bar{\Lambda}_3^k) - \gamma\bigg(\nabla\tilde{f}_{\bar{\lambda}_1}\Delta\bar{\lambda}_1^k + \nabla\tilde{f}_{\bar{\underline{\lambda}}_2}^T\Delta\bar{\underline{\lambda}}_2^k + \mathrm{tr}(\nabla\tilde{f}_{\bar{\Lambda}_3}^T\Delta\bar{\Lambda}_3^k)\bigg)$ **do**

$t^k = \eta^{m_k} t^0$ {line search (backtracking condition)}

end while

$\lambda_1^{k+1} = [\bar{\lambda}_1^k - t^k\nabla\tilde{f}(\bar{\lambda}_1)]_+$, $\underline{\lambda}_2^{k+1} = \bar{\underline{\lambda}}_2^k - t^k\nabla\tilde{f}(\bar{\underline{\lambda}}_2)$, $\Lambda_3^{k+1} = [\bar{\Lambda}_3^k - t^k\nabla\tilde{f}(\bar{\Lambda}_3)]_+$

$a_{k+1} = (1 + \sqrt{4a_k^2 + 1})/2$, and $t^0 = t^k$. {updating the dual variables}

if Duality-Gap $\leq \upsilon$ **then**

break

end if

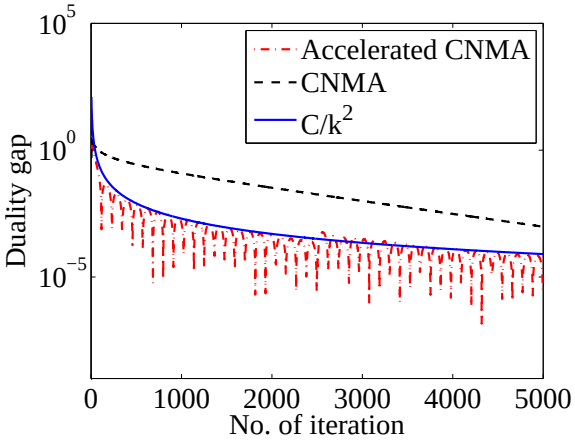
end for

2.5.3.2 Acceleration

The general convergence rate for gradient approach is $\mathcal{O}(\frac{1}{k})$, where k is the iteration number. In [86], it is proved that the extrapolation step makes the convergence faster as much as $\mathcal{O}(\frac{1}{k^2})$. We define the extrapolated solution $\bar{\lambda}^k$ as follows:

$$\bar{\lambda}_1^k = \lambda_1^k + \frac{a_{k-1}-1}{a_k}(\lambda_1^k - \lambda_1^{k-1})$$
$$\bar{\underline{\lambda}}_2^k = \underline{\lambda}_2^k + \frac{a_{k-1}-1}{a_k}(\underline{\lambda}_2^k - \underline{\lambda}_2^{k-1})$$
$$\bar{\Lambda}_3^k = \Lambda_3^k + \frac{a_{k-1}-1}{a_k}(\Lambda_3^k - \Lambda_3^{k-1})$$

where $a_k = \frac{1+\sqrt{4a_{k-1}^2+1}}{2}$. For the pseudo code for the proposed CNMA see Algorithm 3. To illustrate that the proposed acceleration improves the convergence from $\mathcal{O}(1/k)$ to $\mathcal{O}(1/k^2)$, we present a plot of the duality gap vs. the number of iterations for the original CNMA and accelerated CNMA in Fig. 2.3. The evaluation of the SVD in each iteration



(a)

Figure 2.3: Comparison of duality gap for $M = 50$, $L = 80$, $T = 10$, $n = 1000$, $\alpha = 0.1$, and $\beta = 0.01$ for CNMA vs. accelerated CNMA

is expensive and is $\mathcal{O}(\min\{M, L\}^3)$. As in [27, 74, 108], we use the PROPACK package to compute a partial SVD. Because PROPACK can not automatically calculate the singular values which are greater than specific value τ, we use the following procedure. To facilitate the computation of singular value 5 at a time, we set $b_0 = 5$ and update b_{l+1} for $l = 0, 1, \ldots$ as follows:

$$b_{l+1} = \begin{cases} \text{Rank}(\tilde{\Psi}^{k+1}) & \text{if Rank}(\tilde{\Psi}^{k+1}) < b_k \\ \text{Rank}(\tilde{\Psi}^{k+1}) + 5 & \text{if Rank}(\tilde{\Psi}^{k+1}) \geq b_k. \end{cases}$$

This procedure stops when $b_{l+1} = b_l$. Partial SVD calculation reduces the cost of the computation significantly, especially in the low-rank setting. The pseudo code for calculating SVD is in Algorithm 4.

Algorithm 4 SVD calculation using PROPACK

Choose $r_0 = 0$, and $i = 5$
in step l
$b_l = r_{k-1} + 1$
repeat
  $[USV]_{b_l} = \text{SVD}(\Psi'^k)$
  $b_l = b_l + i$
until $s^k_{b_l - i} \leq \frac{1}{\lambda^k_1}$
$r_k = \max\{j : s^k_j > \frac{1}{\lambda^k_1}\}$
$\tilde{\Psi}^{k+1} = \sum_{j=1}^{r_k} (s^k_j - \frac{1}{\lambda^k_1}) u^k_j v^k_j$

2.6 Experimental results

We evaluate both theoretical and computational aspects of the confidence-constrained rank minimization problem. For the theoretical part, we provide the followings: 1) Sensitivity analysis of rank recovery accuracy as a function of ϵ, and 2) Phase diagram

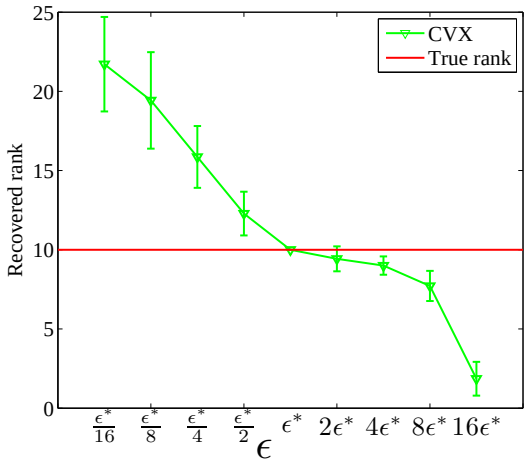
analysis applied to a synthetic dataset to show that the exact rank recovery obtained by CNMA is consistent with the sufficient conditions proposed by Theorem 1. For the computational part, we provide a runtime comparison between CNMA and HDP and show the applicability of CNM for large datasets. For HDP, we use an efficient implementation of the algorithm in Matlab [1] provided by the authors of [107]. Note that in all of our experiments, we fixed the confidence value $1 - \delta_k = 0.9$ and consequently set $k = 3$.

2.6.1 Sensitivity with respect to ϵ

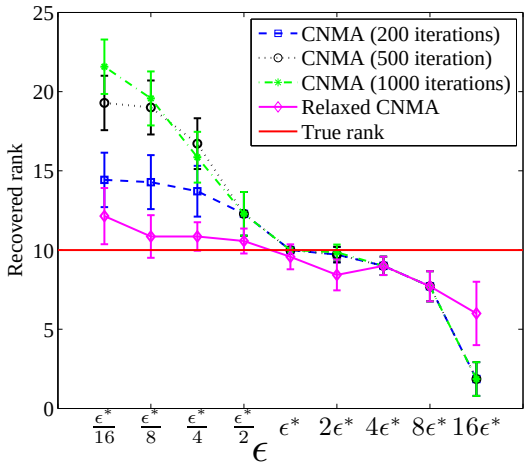
We would like to illustrate the effect of ϵ on rank recovery. Theorem 1 suggests that by selecting $\epsilon = \epsilon^*$ (2.11), rank minimization guarantees exact rank recovery with probability $1 - \delta$. To examine the effect of varying ϵ on rank recovery accuracy, we consider the following setup. We consider a range of values for $\epsilon = [\epsilon^*/16, \epsilon^*/8, \epsilon^*/4, \epsilon^*/2, \epsilon^*, 2\epsilon^*, 4\epsilon^*, 8\epsilon^*, 16\epsilon^*]$. The value of ϵ^* based on (2.11) is equal to 0.2550. We generate matrix Ψ with $M = 50$, $L = 50$, $T = 10$, $\alpha = 0.1$, and $\beta = 0.01$ following the model in Section 2.2.1 and sample $\hat{\Psi}$ 10 times. For each value of ϵ, we solve CNM in (2.14) for each of the ten realization of $\hat{\Psi}$ using `CVX` and CNMA and evaluate the rank of the recovered matrix $\tilde{\Psi}^*$. The rank evaluation is done by counting the number of singular value of matrix $\tilde{\Psi}^*$ exceeding a threshold to avoid miscounting due to numerical errors. The threshold is defined based on the empirical distribution of the smallest nonzero singular values of the true matrix Ψ (i.e., mean minus three times the standard deviation). We compute mean (μ) and standard deviation (σ) of the recovered rank for matrix $\tilde{\Psi}$ and plot the error bar ([mean-std, mean+std]) for both `CVX` and CNMA. Rank estimates as a function of ϵ for

[1] http://www.gatsby.ucl.ac.uk/ ywteh/research/software.html

CVX and for CNMA are shown in Figures 2.4(a) and 2.4(b), respectively. Figures 2.4(a) and 2.4(b) support Theorem 1 by indicating that the choice of $\epsilon = \epsilon^*$ (2.11) leads to exact rank recovery, since for only $\epsilon = \epsilon^*$ the exact rank is recovered for 10 out of 10 leading to $\mu = 10$ and $\sigma = 0$. In other words, as we deviate from ϵ^* the true rank of matrix Ψ can no longer be recovered. We provide the following explanation. When we increase ϵ, the confidence-constrained set may include low-rank matrices which are not in the γ-neighborhood of matrix Ψ. Hence, rank minimization inside the confidence-constrained set may lead to a recovery of a low-rank matrix. On the other hand, as we decrease ϵ the confidence-constrained set may not include the true matrix Ψ. Therefore, the rank of the recovered matrix $\tilde{\Psi}$ may be higher than the rank of matrix Ψ. By comparing Figures 2.4(a) and 2.4(b), we can see that the performance of CNMA is comparable to that of CVX. To assess the effect of the number of CNMA iterations on accuracy, we terminate the algorithm after 200, 500, and 1000 iterations and present the rank recovery results in Figures 2.4(b). Comparing the graphs in Fig. 2.4(b), we observe that with an increased number of iterations the results approach that of CVX. Moreover, CNMA with a smaller number of iterations correctly recovers the rank at $\epsilon = \epsilon^*$. This hints at the potential reduction in computational complexity that CNMA can provide by reducing the number of iterations. For the relaxed CNMA graph in Fig. 2.4(b), we removed the positivity and sum to one constraints to assess the importance of the probability matrix constraints. We observe an increase in variation from the true rank at $\epsilon = \epsilon^*$ (2.11). This suggests that including the probability constraints can improve the rank recovery accuracy.



(a)



(b)

Figure 2.4: This figure shows the sensitivity of rank recovery to the value of ϵ. We scan through a range of values of ϵ and plot the mean of the recovered rank including the confidence intervals for (a) CVX and (b) CNMA.

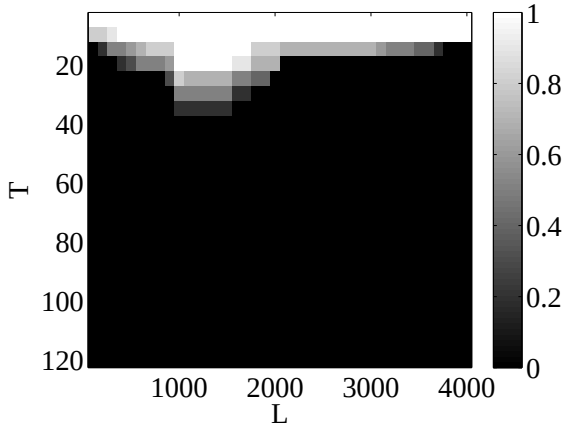
2.6.2 Phase diagram analysis

We use the notion of phase diagram as proposed in [44] to evaluate probability of exact rank recovery using CNMA for a wide range of matrices of different dimensions (i.e., vo-

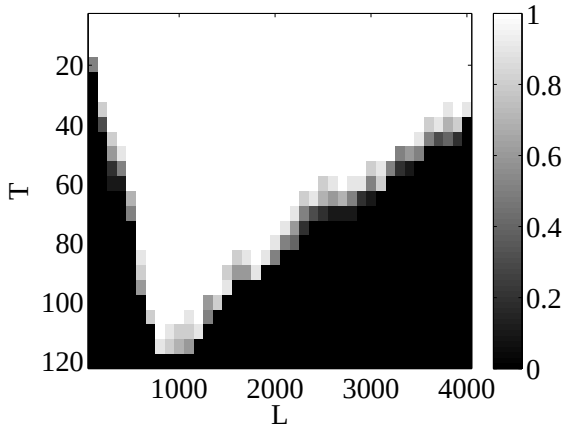
cabulary size terms $\times$ number of documents) and different number of topics and compare it with the sufficient conditions proposed by Theorem 1. We would like to show that the condition proposed in Theorem 1 for rank recovery is still valid when rank minimization is replaced with nuclear norm minimization. We generate $N = 50$ $i.i.d$ realizations of Ψ using the sampling process in Section 2.2.1 with $M = 500$, $n = 1000$, $\alpha = 0.01$, $\beta = 0.001$, over a grid of (L, T), with L ranging through 40 equispaced points in the interval $[100, 4000]$, and T ranging through 24 equispaced points in the interval $[5, 120]$. In Fig. 2.5(a), each pixel intensity corresponds to the empirical estimate of $P(\sigma_T > 2\epsilon)$, i.e., $\sum_{i=1}^{N} I(\sigma_T^{(i)} > 2\epsilon)/N$, where σ_T is the smallest non-zero singular value. To evaluate correct rank recovery probability, for each pixel in phase diagram we produce 20 realization of the pair $(\Psi, \hat{\Psi})$. We run CNMA for each of the 20 realizations of $\hat{\Psi}$ and compared the rank of the recovered matrix $\tilde{\Psi}^*$ with the true rank of matrix Ψ. The rank of matrix $\tilde{\Psi}^*$ is computed following the procedure described in Section 2.6.1. In Fig. 2.5(a), the white area corresponds to success region[2] (the region where the rank recovery is guaranteed with high probability based on Theorem 1). In Fig. 2.5(b), the white area corresponds to exact rank recovery obtained by CNMA. Since the area for exact rank recovery probability obtained by CNMA covers the success region, the sufficient condition proposed by Theorem 1 appear to hold for the heuristic replacement of nuclear norm minimization. Comparing Figures 2.5(a), and 2.5(b) suggests that the sufficient condition for exact rank recovery proposed in Theorem 1 can be further improved. This could be attributed to the fact that the proposed sufficient conditions for exact rank recovery involve several bounds.

The LDA model in Section II depends on two hyperparameters α and β. When α is small the effective number of topics per document is small. Similarly, when β is small the

[2]This notation is used in [44]



(a)



(b)

Figure 2.5: (a) $P(\sigma_T > 2\epsilon)$ for $M = 1000$, $n = 1000$, $\alpha = 0.01$, and $\beta = 0.001$ (b) $\hat{P}$(exact rank recovery) obtained by CNMA.

effective number of words per topic is small. Intuitively, with small α and β the model is simpler (i.e., fewer topics and fewer words per topic). We are interested in evaluating the impact of α and β on the rank recovery rate. In Fig. 2.6, the left hand column shows the phase diagram for exact rank recovery obtained by CNMA for different values of α, and β. As we decrease the value of hyperparameters, the wider area for exact rank

recovery can be covered by CNMA in phase diagram. The middle and left hand side graphs show the singular value scree plot of matrix $\hat{\Psi}$ for the point indicated by darker and lighter pointer on the phase diagram, respectively. The scree plots illustrate the fact that as we decrease α and β, Ψ becomes more distinct, i.e., the gap between the smallest non zero singular value and the following one is more distinguished. Hence, its rank is easier to recover. Moreover, by comparing the scree plots in the middle and left hand columns, it is clear that when the exact rank cannot be recovered by CNMA, the gap in the singular values of matrix $\hat{\Psi}$ cannot be found easily. We would like to emphasize that although the scree plot can be use to study the rank of a matrix, it does not provide a complete solution to the problem, i.e., it fails to suggest an admissible estimate for Ψ. Without probability constraints, an SVD can be use to obtain a low-rank estimate for Ψ. However, in the presence of probability constraint the problem is NP-hard [83].

2.6.3 Computational complexity comparison

We compare the CPU runtime of CNMA with HDP. We consider $(M, L) = [(80, 60)$ $(100, 90)$ $(150, 120)$ $(200, 150)$ $(300, 200)$ $(600, 500)]$. We compute the CPU runtime using a MATLAB built in function {`cputime`}. CNMA and HDP algorithm run on a standard desktop computer with 2.5 GHz CPU (dual core) and 4 GB of memory. Figure 2.7(a) shows the CPU runtime comparison for CNMA vs. HDP. In Fig. 2.7(a), the x-axis shows the dimension of the matrix $L \times M$ and the y-axis shows the elapsed CPU time in seconds. Figure 2.7(a) shows that the runtime of HDP is longer than that of CNMA by at least an order of magnitude. Note that we compared the runtime of `CVX` (using SDPT3 as an SDP solver) with that of CNMA and observed that the runtime of CVX is longer than that of CNMA by over two orders of magnitude. This suggests that CNMA, i.e.,

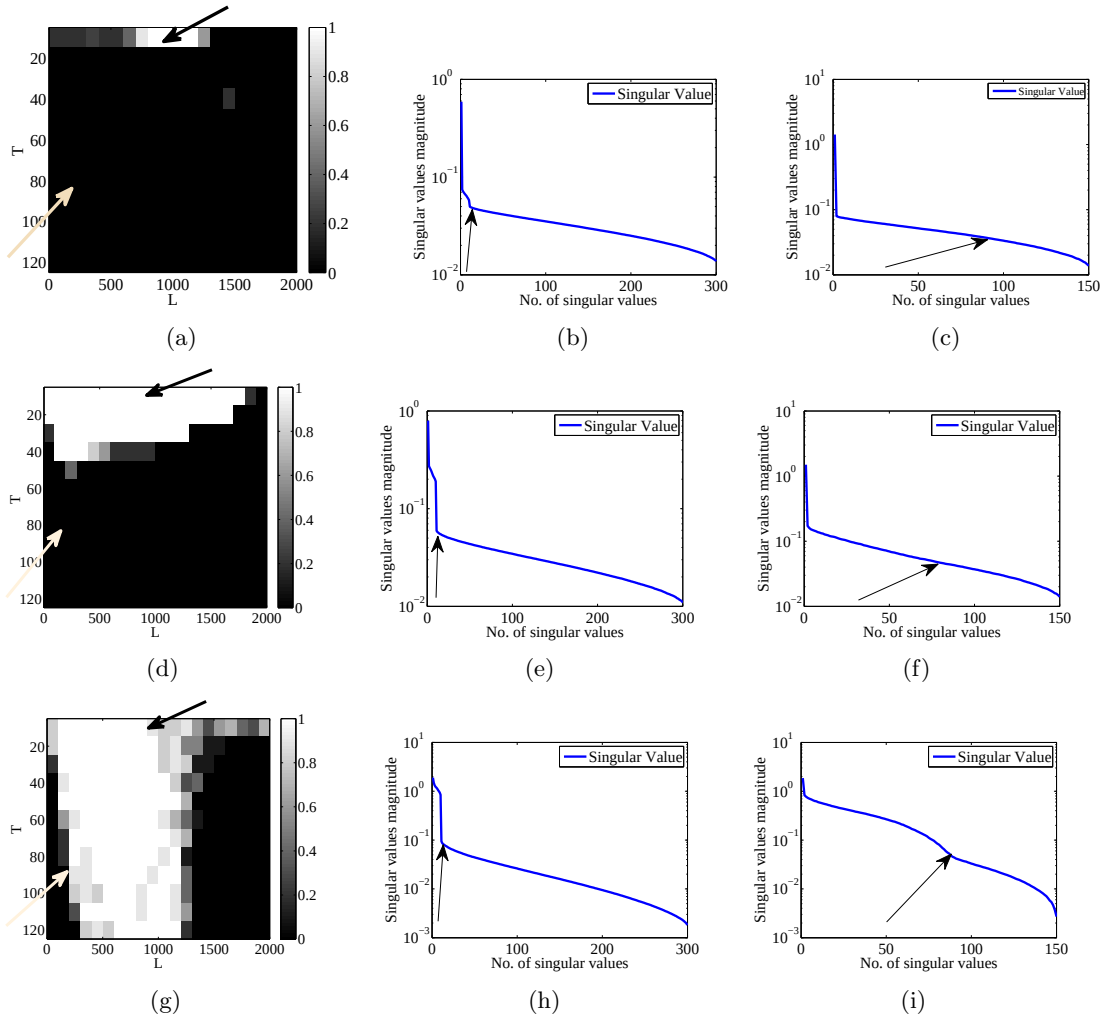
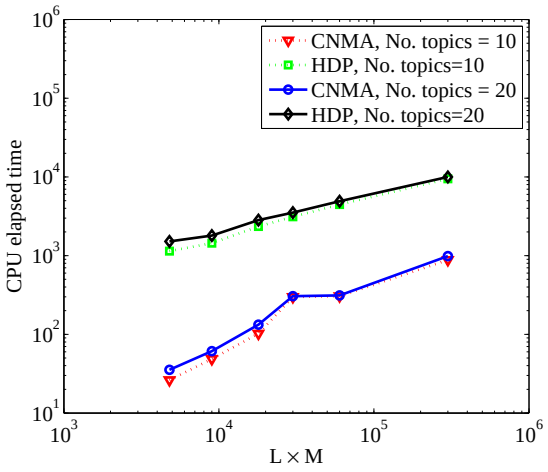


Figure 2.6: This figure shows the effect of the value of the hyperparameters α and β on rank recovery rate. The first column is the phase diagram of $P(\sigma_T > 2\epsilon)$ as a function of the number of topics and the vocabulary size. Each row corresponds to a different setup of the hyperparameters α and β. (a) $\alpha = 1$, $\beta = 1$ (d) $\alpha = 0.5$, and $\beta = 0.1$ (g) $\alpha = 0.1$, and $\beta = 0.01$. The second column is the plot of the singular values for the setting indicated by black arrows. The last column is the plot of the singular values indicated by white arrows. Note that the black arrow in the phase diagram corresponds to the success region proposed by Theorem 1 and the white arrow corresponds to the fail region.

our proposed algorithmic implementing of CNM, provides a fast and feasible solution to practical size problems and diminishes the computational limitations associated with generic solvers.



(a)

Figure 2.7: Runtime comparison between CNMA and HDP.

2.7 Applications

As the previous section suggests, the proposed computationally-efficient algorithmic implementation of CNM can be used to solve problem of realistic dimensions. In this section, we would like to illustrate that the low-rank solution obtained by CNMA provides competitive results to that of LDA, HDP, and the optimal low-rank SVD approximation of matrix $\hat{\Psi}$ in terms of classification accuracy on two real image datasets and three real text datasets.

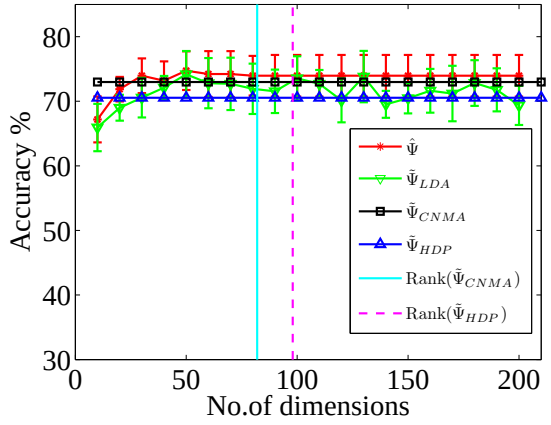
2.7.1 Image datasets

We consider two image datasets MSRCv2[3], and Corel1000[4]. MSRCv2 image dataset contains 591 images in 23 object classes. We perform a multiclass classification for MSRCv2 using the 8 row classes: *'book', 'grass, cow', 'tree, grass, sky', 'bike, building', 'sign', 'water, boat', 'aeroplane, grass, sky', 'road, building'* resulting in a dataset with 240 images in 8 different classes. Corel1000 image dataset contains 1000 images in 10 different classes each includes 100 images. We consider 7 classes: *'buildings', 'buses', 'flowers', 'elephants', 'horses', 'food' and 'mountains'* in our simulation. Note that we excluded the classes which contained images with different format of RGB representations. We randomly sampled 50 images in each class resulting in 350 images in 7 classes.

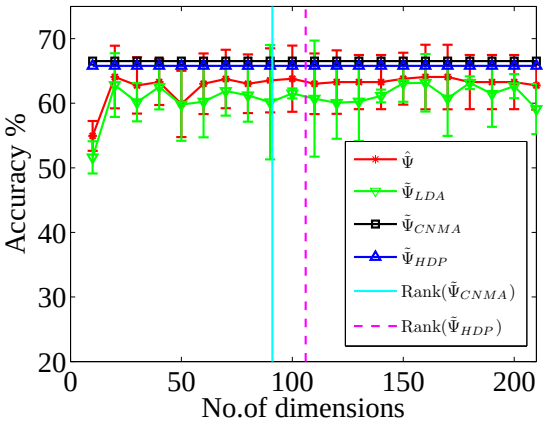
To obtain matrix $\hat{\Psi}$, we take the approach of representing each image as a collection of blocks and mapping each block to a discrete index associated with the closest dictionary template. We separate each image to several $10 \times 10 \times 3$ blocks. To construct the dictionary, we run k-means on the collection of blocks from all images to obtain L cluster centroids. The L centroids are used as the dictionary templates and each block is mapped to the index of the closest dictionary template. We run CNMA, LDA, and HDP to obtain matrix $\tilde{\Psi}^*_{CNMA}, \tilde{\Psi}^*_{LDA}$, and $\tilde{\Psi}^*_{HDP}$, respectively. To find the optimal low-rank approximation of $\hat{\Psi}$, we project the columns of $\hat{\Psi}$ into its top d-largest left singular vectors where d scans through the dimension of matrix $\hat{\Psi}$. We use multi class SVM with Gaussian kernel for classification [32]. Parameters C and γ of SVM model are learned by k-fold cross validation where $k = 5$.

In Figures 2.8 and 2.9, the classification accuracies obtained by running SVM on $\tilde{\Psi}^*_{CNMA}, \tilde{\Psi}^*_{LDA}$, and $\tilde{\Psi}^*_{HDP}$ as well as on different low-rank SVD-based approximations

[3]http://research.microsoft.com/en-us/projects/objectclassrecognition/default.htm
[4]http://wang.ist.psu.edu/docs/related/



(a)



(b)

Figure 2.8: Multiclass classification accuracy for MSRCv2 dataset with number of clusters (a) 200 (b) 500.

of matrix $\hat{\Psi}$ are shown. The classification accuracy provided by matrix $\tilde{\Psi}^*_{CNMA}$ is competitive with that of the others. Since CNMA and HDP determine the number of topics in an automated fashion, the accuracy for each was computed without the need to scan through the different number of topics. The number of dimensions is only relevant for the LDA and SVD approaches, in which the number of topics is an additional input

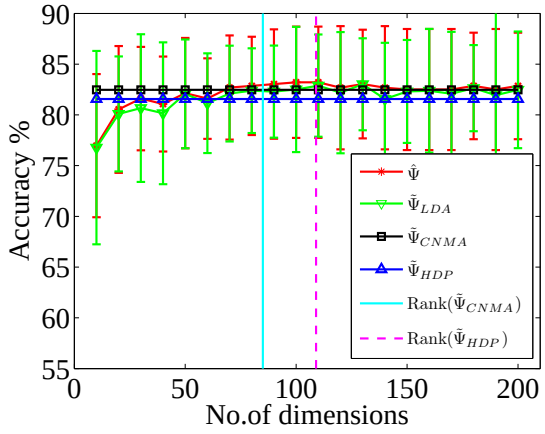
to the algorithm. In both Figures 2.8 and 2.9, the vertical line shows the rank of the recovered matrix $\tilde{\Psi}^*$. We observe that the classification accuracy for the SVD based dimension reduced $\hat{\Psi}$ remains stable for ranks greater than Rank(Ψ^*). This suggests that the number of rank proposed by CNMA can be considered for dimension reduction of matrix $\hat{\Psi}$. Moreover, $\tilde{\Psi}^*_{CNMA}$ produces competitive performance results to that of $\tilde{\Psi}^*_{LDA}$ and $\tilde{\Psi}^*_{HDP}$.

In [71], supervised LDA was run on MSRCv2 dataset. The highest classification accuracy obtained by running variational Bayes on LDA in [71] is 69%, which is 5% percent below the results obtained by CNMA. We have to emphasize that since CNM is an unsupervised approach for dimension reduction, its classification accuracy can be further improved by introducing class label information to CNM. We also ran similar simulations using the SIFT representation of the features proposed by [76] instead of blocks. The sparsity of matrix $\hat{\Psi}$ obtained by SIFT representation is lower than the sparsity of $\hat{\Psi}$ obtained using a block representation. The theory we present in this section and the numerical evaluations in Section 2.6.2 suggest that when α and β are large (lower sparsity), the rank recovery success region is diminished. This is consistent with the decrease in performance we observed.

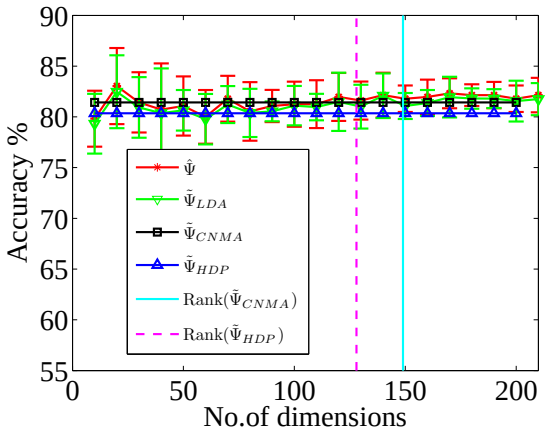
2.7.2 Text datasets

We evaluate the classification accuracy of the proposed CNMA approach with HDP, LDA and SVD approaches on TDT2[5], Reuters[6], and 20Newsgroup[7] datasets. The TDT2 corpus consists of data collected during the first half of 1998 and taken from 6 sources

[5]http://www.nist.gov/speech/tests/tdt/tdt98/index.htm
[6]http://www.daviddlewis.com/resources/testcollections/reuters21578/
[7]http://people.csail.mit.edu/jrennie/20Newsgroups/



(a)



(b)

Figure 2.9: Multiclass classification accuracy for Corel1000 dataset with number of clusters (a) 200 (b) 500.

including 2 newswires (APW, NYT), 2 radio programs (VOA, PRI), and 2 television programs (CNN, ABC), total 11201 documents in 96 different categories. The 20 Newsgroups dataset is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. Reuters-21578 corpus contains 21578 documents in 135 categories. We use here the ModApte version of the Reuters dataset.

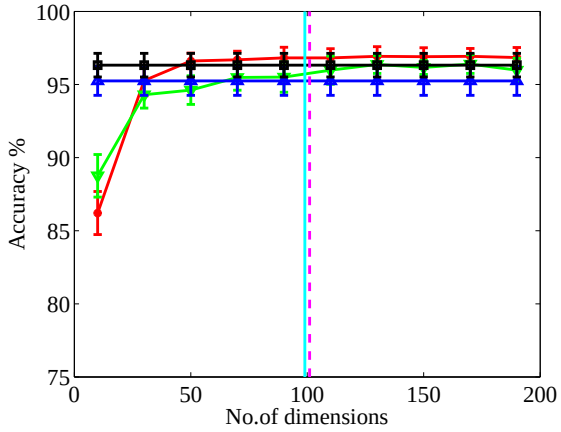
Documents with multiple category labels are discarded leaving 8293 documents in 65 categories. In our experiments we removed documents with low number of words. Table 2.2 shows the summary of each dataset that we use in our analysis. We compare

Table 2.2: Text Dataset summary

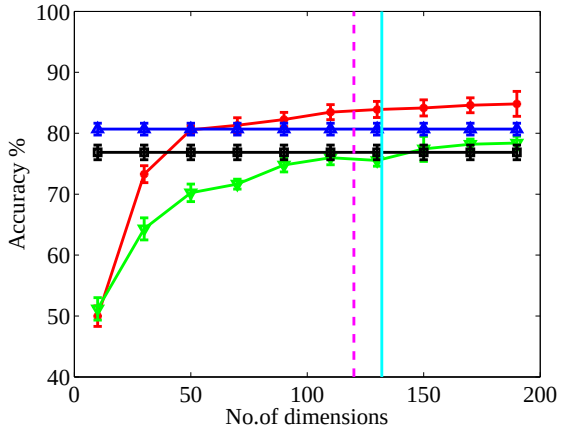
	TDT2	20Newsgroup	Reuters
No. of documents	3807	4342	3228
Vocabulary size	4350	4612	3071
No. of category	30	20	10
Minimum no. of words per document (n_d)	180	150	50

CNMA with HDP, LDA, and low-rank SVD approximation of matrix $\hat{\Psi}$. We use multi-class liblinear SVM[8], which is well suited for document classification. We use 5-fold cross validation to optimize the parameter C of the SVM algorithm. Figure 2.10 shows the results of classification for different datasets. We omitted the legend of Fig. 2.10(a) and Fig. 2.10(b) which are identical to the legend of Fig. 2.10(c). By comparing the results in Fig. 2.10, we observe that the performance of CNMA is competitive with HDP, LDA, and SVD. Moreover, the number of topics found by both CNMA and HDP algorithms is quite similar. This suggests that the dimension of the latent space discovered by HDP can be recovered by CNMA as well.

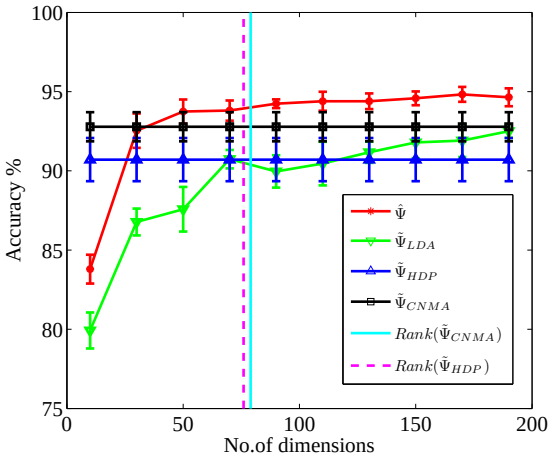
[8]http://www.csie.ntu.edu.tw/cjlin/liblinear/



(a)



(b)



(c)

Figure 2.10: Classification accuracy for (a) TDT2, b) 20Newsgroup, and (c) Reuters

2.8 Appendix

2.8.1 Derivative of $\frac{\lambda_1}{2}\|D_{\frac{1}{\lambda_1}}(\Psi')\|_F^2$ with respect to λ_1

The derivative of $\frac{\lambda_1}{2}\|D_{\frac{1}{\lambda_1}}(\Psi')\|_F^2$ with respect to λ_1 is

$$\frac{d\frac{\lambda_1}{2}\|D_{\frac{1}{\lambda_1}}(\Psi')\|_F^2}{d\lambda_1} = \frac{1}{2}\|D_{\frac{1}{\lambda_1}}(\Psi')\|_F^2 + \frac{1}{\lambda_1}\|D_{\frac{1}{\lambda_1}}(\Psi')\|_* - \frac{1}{\lambda_1}\text{tr}((1\underline{\lambda}_2^2 + \Lambda_3)^T D_{\frac{1}{\lambda_1}}(\Psi'))) \quad (2.18)$$

Proof:

Using the product rule, the derivative of $\frac{\lambda_1}{2}\|D_{\frac{1}{\lambda_1}}(\Psi')\|_F^2$ with respect to λ_1 can be expressed as:

$$\frac{d\frac{\lambda_1}{2}\|D_{\frac{1}{\lambda_1}}(\Psi')\|_F^2}{d\lambda_1} = \frac{1}{2}\|D_{\frac{1}{\lambda_1}}(\Psi')\|_F^2 + \frac{\lambda_1}{2}\frac{d\|D_{\frac{1}{\lambda_1}}(\Psi')\|_F^2}{d\lambda_1}. \quad (2.19)$$

Since $D_{\frac{1}{\lambda_1}}(\Psi') = U(S - \frac{1}{\lambda_1}I)_+V^T$, we have $\|D_{\frac{1}{\lambda_1}}(\Psi')\|_F^2 = \text{tr}\left(D_{\frac{1}{\lambda_1}}(\Psi')^T D_{\frac{1}{\lambda_1}}(\Psi')\right) = \text{tr}\left((S - \frac{1}{\lambda_1}I)_+^2\right)$. Therefore, the second term on the RHS of (2.19) is

$$\begin{aligned}
\frac{\lambda_1}{2}\frac{d}{d\lambda_1}(\|D_{\frac{1}{\lambda_1}(\Psi')}\|_F^2) &= \frac{\lambda_1}{2}\frac{d}{d\lambda_1}\text{tr}\left((S - \frac{1}{\lambda_1}I)_+^2\right) \\
&= \lambda_1\text{tr}\left(\frac{d(S - \frac{1}{\lambda_1}I)}{d\lambda_1}(S - \frac{1}{\lambda_1}I)_+\right) \\
&= \lambda_1\text{tr}\left(\frac{dS}{d\lambda_1}(S - \frac{1}{\lambda_1}I)_+\right) + \frac{1}{\lambda_1}\text{tr}\left((S - \frac{1}{\lambda_1}I)_+\right) \\
&= \lambda_1\text{tr}\left(\frac{dS}{d\lambda_1}(S - \frac{1}{\lambda_1}I)_+\right) + \frac{1}{\lambda_1}\|D_{\frac{1}{\lambda_1}}(\Psi')\|_*. \quad (2.20)
\end{aligned}$$

Since $\text{tr}\left(\frac{dS}{d\lambda_1}(S - \frac{1}{\lambda_1}I)_+\right) = \text{tr}\left((\frac{d\Psi'}{d\lambda_1})^T D_{\frac{1}{\lambda_1}}(\Psi')\right)$ [90], we have $\lambda_1\text{tr}\left(\frac{dS}{d\lambda_1}(S - \frac{1}{\lambda_1}I)_+\right) =$

$-\frac{1}{\lambda_1}\text{tr}((1\underline{\lambda}_2^T + \Lambda_3)^T D_{\frac{1}{\lambda_1}}(\Psi')))$ and consequently

$$\frac{\lambda_1}{2}\frac{d\|D_{\frac{1}{\lambda_1}}(\Psi')\|_F^2}{d\lambda_1} = -\frac{1}{\lambda_1}\text{tr}((1\underline{\lambda}_2^T + \Lambda_3)^T D_{\frac{1}{\lambda_1}}(\Psi'))) + \frac{1}{\lambda_1}\|D_{\frac{1}{\lambda_1}}(\Psi')\|_*. \tag{2.21}$$

Substituting (2.21) into (2.19), we obtain (2.18).

2.8.2 Proof of probability bound for estimation error

To prove the probability bound for the estimation error of rank recovery in CRM, we defined two random quantities $Q = \sum_{d=1}^{M} n_d Q_d$ and $Q' = \sum_{d=1}^{M} Q_d$, where $Q_d = \sum_{l=1}^{L}(\Psi_{ld} - \hat{\Psi}_{ld})^2$. We use the one-tailed Chebyshev's inequality for random variable X as following:

$$P\left(X \geq E(X) + k\sqrt{Var(X)}\right) \leq \frac{1}{1+k^2}. \tag{2.22}$$

To compute the Chebyshev bound, we need to evaluate mean and variance of random quantity Q_d. First we start with calculation of the expected value of random variable Q_d.

$$\begin{aligned} E(Q_d) &= \sum_{l=1}^{L} E(\hat{\Psi}_{ld} - \Psi_{ld})^2 \\ &= \text{Var}(\hat{\Psi}_{ld}) = \sum_{l=1}^{L} \frac{\Psi_{ld}(1-\Psi_{ld})}{n_d} \\ &= \frac{1}{n_d}(1 - \sum_{l=1}^{L} \Psi_{ld}^2) \end{aligned} \tag{2.23}$$

Note that $\text{Var}(\hat{\Psi}_d) = \frac{\Psi_{ld}(1-\Psi_{ld})}{n_d}$.

2.8.2.1 $Var(Q_d)$

The variance of Q_d can be calculated as follows (for notational ease we define $I_{ij} = I(X_i = j)$):

$$\begin{aligned}\mathrm{Var}(Q_d) &= \sum_{l=1}^{L}\sum_{m=1}^{L}\Bigg(E\Bigg[\Big(\frac{1}{n_d}\sum_{i=1}^{n_d} I_{il} - \Psi_{ld}\Big)^2\Big(\frac{1}{n_d}\sum_{j=1}^{n_d} I_{jm} - \Psi_{md}\Big)^2\Bigg] - \\ &\quad E\Bigg[\Big(\frac{1}{n_d}\sum_{i=1}^{n_d} I_{il} - \Psi_{ld}\Big)^2\Bigg]E\Bigg[\Big(\frac{1}{n_d}\sum_{j=1}^{n_d} I_{jm} - \Psi_{md}\Big)^2\Bigg]\Bigg) \qquad (2.24)\end{aligned}$$

We compute the second term on the RHS of (2.24) as follows:

$$E\Bigg[\Big(\frac{1}{n_d}\sum_{i=1}^{n_d} I_{il} - \Psi_{ld}\Big)^2\Bigg]E\Bigg[\Big(\frac{1}{n_d}\sum_{i=1}^{n_d} I_{jm} - \Psi_{md}\Big)^2\Bigg] = \frac{\Psi_{ld}(1-\Psi_{ld})}{n_d} \times \frac{\Psi_{md}(1-\Psi_{md})}{n_d}$$

For the first term on the RHS of (2.24), we have:

$$\begin{aligned}&E\Bigg[\Big(\frac{1}{n_d}\sum_{i=1}^{n_d} I_{il} - \Psi_{ld}\Big)^2\Big(\frac{1}{n_d}\sum_{j=1}^{n_d} I_{jm} - \Psi_{md}\Big)^2\Bigg] = \\ &\frac{1}{n_d^4}\Bigg(\sum_i\sum_j\sum_k\sum_t E\Bigg[\Big(I_{il} - \Psi_{ld}\Big)\Big(I_{jl} - \Psi_{ld}\Big)\Big(I_{km} - \Psi_{md}\Big)\Big(I_{tm} - \Psi_{md}\Big)\Bigg]\Bigg)\end{aligned}$$

To evaluate $E\left[\left(I_{il} - \Psi_{ld}\right)\left(I_{jl} - \Psi_{ld}\right)\left(I_{km} - \Psi_{md}\right)\left(I_{tm} - \Psi_{md}\right)\right]$, we consider all the alternatives of i, j, k, l as follows (the enumeration of each alternative is specified in the bracket):

1. $[n_d]$ $i = j = k = t$

$$\left(I_{il} - \Psi_{ld}\right)^2 = I_{il}\left(1 - 2\Psi_{ld}\right) + \Psi_{ld}^2$$

$$E\left[\left(I_{il}\left(1 - 2\Psi_{ld}\right) + \Psi_{ld}^2\right)\left(I_{im}\left(1 - 2\Psi_{md}\right) + \Psi_{md}^2\right)\right] =$$

$$\delta_{lm}\Psi_{ld}\left(1-2\Psi_{ld}\right)^2+\Psi_{ld}\left(1-2\Psi_{ld}\right)\Psi_{md}^2+\Psi_{ld}^2\Psi_{md}\left(1-2\Psi_{md}\right)+\Psi_{ld}^2\Psi_{md}^2$$

2. $[4n_d(n_d-1)]$ $(i=j=k\neq t, i=j=t\neq k, i=k=t\neq j, j=k=t\neq i)$

$$E\left[\left(I_{il}-\Psi_{ld}\right)^2\left(I_{im}-\Psi_{md}\right)\left(I_{tm}-\Psi_{md}\right)\right]=0$$

3. $[n_d(n_d-1)]$ $i=j\neq k=t$

$$E\left[\left(I_{il}-\Psi_{ld}\right)^2\right]E\left[\left(I_{jm}-\Psi_{md}\right)^2\right]=\frac{\Psi_{ld}(1-\Psi_{ld})}{n_d}\times\frac{\Psi_{md}(1-\Psi_{md})}{n_d}$$

4. $[2n_d(n_d-1)]$ $(i=k\neq j=t, i=t\neq j=k)$

$$2E\left[\left(I_{il}-\Psi_{ld}\right)\left(I_{jm}-\Psi_{md}\right)\right]^2=2\left[\delta_{lm}\Psi_{ld}-\Psi_{ld}\Psi_{md}-\Psi_{ld}\Psi_{md}+\Psi_{ld}\Psi_{md}\right]^2$$
$$=2\left(\delta_{lm}\Psi_{ld}-\Psi_{ld}\Psi_{md}\right)^2=2\left(\delta_{lm}\Psi_{ld}^2\left(1-2\Psi_{ld}\right)+\Psi_{ld}^2\Psi_{md}^2\right)$$

5. $[6n_d(n_d-1)(n_d-2)]$ $(i=j\neq k\neq t,$ and all the combinations of 3 out of 4)

$$E\left[\left(I_{il}-\Psi_{ld}\right)^2\left(I_{km}-\Psi_{md}\right)\left(I_{tm}-\Psi_{md}\right)\right]=0$$

6. $[n_d(n_d-1)(n_d-2)(n_d-3)]$ $i\neq j\neq k\neq t$

$$E\left[\left(I_{il}-\Psi_{ld}\right)\left(I_{jl}-\Psi_{ld}\right)\left(I_{km}-\Psi_{md}\right)\left(I_{tm}-\Psi_{md}\right)\right]=0$$

By adding all the alternatives from one to six and organizing them, we get the following expression for $Var(Q_d)$:

$$
\begin{aligned}
&Var(Q_d) = \frac{2}{n_d^2}\sum_{l=1}^{L}\sum_{m=1}^{L}\left(\delta_{lm}\Psi_{ld}^2\left(1-2\Psi_{ld}\right)+\Psi_{ld}^2\Psi_{md}^2\right)+\\
&\frac{1}{n_d^3}\sum_{l=1}^{L}\sum_{m=1}^{L}\Bigg(\delta_{lm}\Psi_{ld}\left(1-2\Psi_{ld}\right)^2+\Psi_{ld}\left(1-2\Psi_{ld}\right)\Psi_{md}^2+\Psi_{ld}^2\Psi_{md}\left(1-2\Psi_{md}\right)+\Psi_{ld}^2\Psi_{md}^2\\
&-\Psi_{ld}(1-\Psi_{ld})\Psi_{md}(1-\Psi_{md})-2\left(\delta_{lm}\Psi_{ld}^2(1-2\Psi_{md})+\Psi_{ld}^2\Psi_{md}^2\right)\Bigg)\\
&=\frac{2}{n_d^2}\Bigg(\sum_{l=1}^{L}\Psi_{ld}^2-2\sum_{l=1}^{L}\Psi_{ld}^3+(\sum_{l=1}^{L}\Psi_{ld}^2)^2\Bigg)\\
&+\frac{1}{n_d^3}\Bigg(8\sum_{l=1}^{L}\Psi_{ld}^3-6(\sum_{l=1}^{L}\Psi_{ld}^2)^2-2\sum_{l=1}^{L}\Psi_{ld}^2\Bigg) \qquad (2.25)
\end{aligned}
$$

The first component on RHS of (2.25) can be bounded using Cauchy-Schwartz as $\left(\sum\Psi_{ld}^{1.5}\Psi_{ld}^{0.5}\right)^2\leq\sum_l\Psi_{ld}^3\sum_l\Psi_{ld}$. Hence, $\left(\sum_l\Psi_{ld}^2\right)^2\leq\sum_l\Psi_{ld}^3$. Thus,

$$
\begin{aligned}
&\frac{2}{n_d^2}\Bigg(\sum_{l=1}^{L}\Psi_{ld}^2-2\sum_{l=1}^{L}\Psi_{ld}^3+(\sum_{l=1}^{L}\Psi_{ld}^2)^2\Bigg)\leq\frac{2}{n_d^2}\Bigg(\sum_{l=1}^{L}\Psi_{ld}^2-2(\sum_{l=1}^{L}\Psi_{ld}^2)^2+(\sum_{l=1}^{L}\Psi_{ld}^2)^2\Bigg)\\
&=\frac{2}{n_d^2}\Bigg(\sum_{l=1}^{L}\Psi_{ld}^2-(\sum_{l=1}^{L}\Psi_{ld}^2)^2\Bigg)=\frac{2}{n_d^2}(t-t^2)=\frac{2}{n_d^2}(1/4-(t-1/2)^2)\leq\frac{1}{2n_d^2},
\end{aligned}
$$

where $t=\sum_{l=1}^{L}\Psi_{ld}^2$. For the second component term on RHS of (2.25) since $\sum_l\Psi_{ld}^3\leq\sum_l\Psi_{ld}^2$, we have

$$
\begin{aligned}
&\frac{1}{n_d^3}\Bigg(8\sum_{l=1}^{L}\Psi_{ld}^3-6(\sum_{l=1}^{L}\Psi_{ld}^2)^2-2\sum_{l=1}^{L}\Psi_{ld}^2\Bigg)\leq\frac{6}{n_d^3}\Bigg(\sum_{l=1}^{L}\Psi_{ld}^2-(\sum_{l=1}^{L}\Psi_{ld}^2)^2\Bigg)\\
&=\frac{6}{n_d^3}(1/4-(t-1/2)^2)\leq\frac{3}{2n_d^3}.
\end{aligned}
$$

The mean of Q and Q' can be bounded as follows:

$$E(Q) = \sum_{d=1}^{M} n_d E(Q_d) = M - \sum_{d=1}^{M}\sum_{l=1}^{L} \Psi_{ld}^2 \leq M,$$
$$E(Q') = \sum_{d=1}^{M} E(Q_d) = \sum_{d=1}^{M} \frac{1}{n_d} - \sum_{d=1}^{M}\sum_{l=1}^{L} \Psi_{ld}^2 \leq \sum_{d=1}^{M} \frac{1}{n_d}, \tag{2.26}$$

since $-\sum_{d=1}^{M}\sum_{l=1}^{L} \Psi_{ld}^2 \leq 0$. Note that Q_d, $d = 1, \ldots, M$ are *i.i.d.* random variables, thus the variance of Q and Q' can be computed as the sum of variance of Q_d.

$$Var(Q) = \sum_{d=1}^{M} n_d^2 Var(Q_d) \leq \frac{M}{2} + \frac{3}{2}\sum_{d=1}^{M} \frac{1}{n_d}$$
$$Var(Q') = \sum_{d=1}^{M} Var(Q_d) \leq \sum_{d=1}^{M} \frac{1}{2n_d^2} + \sum_{d=1}^{M} \frac{3}{2n_d^3}. \tag{2.27}$$

Using the one-tailed Chebyshev inequality, we have the following probability bound for Q and Q':

$$P\left(Q \geq M + k\sqrt{\frac{M}{2}(1 + 3/M\sum_{d=1}^{M}\frac{1}{n_d})}\right) \leq \frac{1}{1+k^2},$$
$$P\left(Q' \geq \sum_{d=1}^{M}\frac{1}{n_d} + k\sqrt{(\sum_{d=1}^{M}\frac{1}{2n_d^2} + \sum_{d=1}^{M}\frac{3}{2n_d^3})}\right) \leq \frac{1}{1+k^2}.$$

Alternatively, we say w.p. $1-\delta_k$, ,$\delta_k = \frac{1}{1+k^2}$, we have $Q = \sum_{d=1}^{M}\sum_{l=1}^{L} n_d\left(\hat{\Psi}_{ld} - \Psi_{ld}\right)^2 \leq \epsilon^2(\delta_k)$, where

$$\epsilon^2(\delta_k) = \epsilon^{*2}(\delta_k) = M + k\sqrt{\frac{M}{2}\left(1 + 3/M\sum_{d=1}^{M}\frac{1}{n_d}\right)},$$

and $Q' = \sum_{d=1}^{M} \sum_{l=1}^{L} \left(\hat{\Psi}_{ld} - \Psi_{ld} \right)^2 \leq \epsilon'^2(\delta_k)$, where

$$\epsilon'^2(\delta_k) = \epsilon'^{*2}(\delta_k) = \sum_{d=1}^{M} \frac{1}{n_d} + k \sqrt{\left(\sum_{d=1}^{M} \frac{1}{2n_d^2} + \sum_{d=1}^{M} \frac{3}{2n_d^3} \right)}.$$

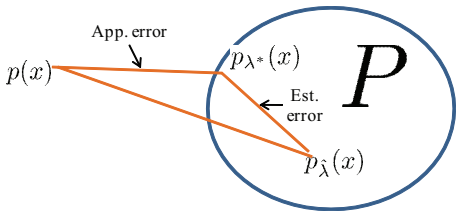
Chapter 3: Entropy Estimation Using the Principle of Maximum Entropy

3.1 Introduction

Information theory quantities such as entropy and mutual information are widely used in data analysis, signal processing, and machine learning. When an underlying model for data is unavailable, sample-based entropy estimation is required. Entropy estimation has been applied in anomaly detection, image segmentation, estimation of manifold dimension and feature selection (e.g., [88]). We consider the estimation of the entropy of a continuous random variable characterized by a PDF. In the discrete case, raw counts are used to estimate the probability for each discrete value and consequently, entropy is estimated using the plug-in method. In the continuous case, two main approaches exist. In the first approach, the PDF is approximated and then the result of the approximation is plugged into the entropy formula (e.g., *kernel density*, *histogram*). In the second approach, the entropy is estimated directly from samples (e.g., *sample spacing*, *nearest neighbors*, and *entropic spanning graph*, see [16] for a review).

The main contribution of this section is developing a new entropy estimator based on the principle of maximum entropy and greedy m-term approximation. We also provide the analysis of the estimation error, specifically an in probability error bound in terms of the problem parameters (e.g., number of samples, number of the approximation terms). The error of the proposed estimator is $\mathcal{O}(\sqrt{\log n/n})$; only a factor of $\sqrt{\log n}$ away from

the classical statistical parameter estimation error $\mathcal{O}(\sqrt{1/n})$. Using numerical examples, we demonstrate the superiority of our algorithm as compared with the other well known algorithms.



(a)

Figure 3.1: Maximum entropy approach for approximating $p(x)$ with $p^*_\lambda(x)$ and estimating with $p_{\hat{\lambda}}(x)$

3.2 Problem formulation

We consider the estimation of the entropy of random variable X from n *i.i.d.* samples of it. Let X be a random variable with a PDF $p(x)$. The entropy of X is given by

$$H(p) = E_p[-\log p(x)] = -\int p(x) \log p(x) dx. \tag{3.1}$$

We are interested in an entropy estimator $\hat{H} : \mathcal{X}^n \to \mathbb{R}$ of $H(p)$, which takes $x_1, x_2, \ldots, x_n \in \mathcal{X}$ as the input. We seek a consistent estimator in the following sense:

$$\lim_{n \longrightarrow \infty} \hat{H}_n(x_1, \ldots, x_n) \longrightarrow H(p) \text{ in probability.} \tag{3.2}$$

We are also interested in quantification of the estimation error $H(p) - \hat{H}(p)$.

3.3 Solution framework

To estimate the entropy given in (3.1), two approximations are typically considered. The first involves replacing the expectation with a sample average. The second involves the more challenging task of estimating $p(x)$ or $\log p(x)$. To address the second approximation, we consider a maximum entropy approach to model $p(x)$.

3.3.1 Maximum entropy framework for entropy estimation

Assume that you have access to the expected value of m different features $\{\phi_j(x)\}_{j=1}^m$ (e.g., mean $E[x]$ and second-order moment $E[x^2]$) w.r.t to PDF $p(x)$. Even if m is large, one cannot identify $p(x)$ uniquely. Maximum entropy principle allows for finding a unique distribution among all distributions that satisfy a set of constraints:

$$\max_p H(p) \quad s.t. \quad E_p[\phi_j(x)] = \alpha_j, \quad j = 1, 2, \ldots, m, \tag{3.3}$$

where $H(p)$ is given in (3.1), $\phi_j(x)$ is a feature function, and α_j is the expected value of the jth feature. The distribution that solves the constrained maximization in (3.3) is given by

$$p_\lambda(x) = \exp(\sum_{j=1}^m \lambda_j \phi_i(x) - Z(\lambda)), \tag{3.4}$$

where $\lambda \in \mathbb{R}^m$ is the solution to the set of equations $E_{p_\lambda}[\phi_j(x)] = \alpha_j$ for $j = 1, 2, \ldots, m$ and $Z(\lambda) = \log \int \exp(\sum_{j=1}^m \lambda_j \phi_j(x)) dx$. Substituting the PDF given by (3.4) into (3.1),

yields a parametric expression for the entropy:

$$H(p_\lambda) = Z(\lambda) - \sum_{i=1}^{m} \lambda_j E_{p_\lambda}[\phi_j(x)]. \tag{3.5}$$

The set of PDFs $\mathcal{P} = \{p_\lambda | \lambda \in \mathbb{R}^m\}$ provides an approximation space for p. The set $\mathcal{P}$ is convex [38] and as a results, a unique $p_\lambda^* \in \mathcal{P}$ can be found, which minimizes the Kullback-Leibler (KL) divergence between the distribution p and its approximation p_λ given by $D(p||p_\lambda) = \int p(x) \log(p(x)/p_\lambda(x))dx$ (see illustration in Fig. 3.1(a)). Such p_{λ^*} satisfies $E_p[\phi_j] = E_{p_{\lambda^*}}[\phi_j]$ for $j = 1, 2, \ldots, m$. The entropy of p_{λ^*} is given by

$$H(p_{\lambda^*}) = \min_\lambda Z(\lambda) - \sum_{j=1}^{m} \lambda_j E_p[\phi_j(x)]. \tag{3.6}$$

Barron *et. al.* showed that under certain conditions, there exists a choice of m ϕ_j's allowing for an accurate approximation of $p(x)$, i.e., $0 \leq D_{kl}(p\|p_{\lambda^*}) = H(p_{\lambda^*}) - H(p) \leq c/m$ [8]. This approximation capability of the maximum entropy framework is key to our method suggesting the idea of replacing $H(p)$ with $H(p_\lambda^*)$.

Since only observations $x_1, x_2, \ldots, x_n$ from $p(x)$ are available, one cannot obtain λ^* based on $p(x)$. Instead, $\hat{\lambda}$ is obtained by maximizing the likelihood or equivalently by minimizing the negative log-likelihood $Z(\lambda) - \sum_{j=1}^{m} \lambda_j E_{\hat{p}}[\phi_j(x)]$, where $\hat{p}$ is the empirical distribution for which $E_{\hat{p}}[f(x)] = \frac{1}{n}\sum_{i=1}^{n} f(x_i)$. The entropy of $p_{\hat{\lambda}}$ is given by

$$H(p_\lambda) = \min_\lambda Z(\lambda) - \sum_{j=1}^{m} \lambda_j E_{\hat{p}}[\phi_j(x)]. \tag{3.7}$$

The sample-based entropy estimated in (3.7) provides an estimate to (3.6). By concentration of measure, i.e., the property that $E_{\hat{p}}[f(x)] \to E_p[f(x)]$ one can show that

(3.7) converges to (3.6) in probability. Motivated by the approximation and estimation capabilities of the framework, we proceed with the description of two specific estimators and their properties.

3.3.2 Proposed estimators

There are two key issues which have to be addressed in finding an optimum estimator for the entropy using the framework of maximum entropy. The first issue is to find the optimum λ in (3.7) which can be done by a variety of convex optimization tools. Specifically for this model, iterative scaling is a common approach [17]. The second issue is to find the best set of ϕ's which provides an accurate approximation for the true entropy. For that end, we define a collection of feature functions ϕ given by $\Phi = \{\phi_\theta | \theta \in \Theta\}$ with $\Theta \subseteq \mathbb{R}^d$. Suppose $\phi_{\theta_1}, \ldots, \phi_{\theta_m}$ are the features used to approximate $p(x)$. Thus, the entropy estimator is:

$$\hat{H}^{(m)}(\theta_1, \ldots, \theta_m) = \min_{\lambda} Z(\lambda; \theta) - \sum_{l=1}^{m} \lambda_l E_{\hat{p}}[\phi_{\theta_l}(x)]. \tag{3.8}$$

While the solution to λ is straightforward following the maximum entropy approach, the choice of θ is not trivial. Two estimators are proposed to address the selection of θ in (3.8) and analysis of the error is provided.

3.3.2.1 Brute-force m-term entropy estimator

We propose the following estimator

$$\hat{H}_1^{(m)} = \min_{\theta_1,\ldots,\theta_m,\lambda_1,\ldots,\lambda_m} Z(\lambda;\theta) - \sum \lambda_l E_{\hat{p}}[\phi_{\theta_l}(x)]. \tag{3.9}$$

The solution to (3.9) presents a strategy for finding the ϕ_j in (3.7). The joint minimization of $\theta_1,\ldots,\theta_m$ presents a computational challenge. However, the estimator performance can allow us to understand the limitations of the approach.

Theorem 6 *Let* $\tilde{H}_1^{(m)} = \hat{H}_1^{(m)} - C/2m$. *The estimation error associated with* $\tilde{H}_1^{(m)}$ *satisfies:,*

$$|\tilde{H}_1^{(m)} - H(p)| \leq \frac{C}{2m} + \frac{ML}{\sqrt{n}}\sqrt{2(\log\frac{2m}{\delta})} \tag{3.10}$$

with probability at least $1-\delta$, *where* $C = \frac{1}{2}e^{||\log p - \log p_{\lambda^*}||_\infty}$ $||\log p||_{\mathcal{L}_1}$, $||\phi_\theta||_\infty \leq M$, *and* $||\lambda||_1 \leq L$.

Theorem 6 decomposes the error of estimating the entropy into two parts: approximation error and estimation error (analogous to the familiar bias and variance decomposition in classical statistics). The first term on the RHS is corresponding to approximation error. Increasing the number of terms m provides a rich basis for the space that includes the target function $\log p(x)$ and hence reduce the error. Simultaneously, the estimation error is increased. The second term is the estimation error which decreases as the number of samples n increases. Constant C depends on the $||f||_\infty$ where $f(x) = \log p(x)$. Common in approximation theory, the approximate function $f(x)$ is assumed to be bounded $||f||_\infty \leq M$. The details of the derivation of parameter C are given in [8]. Due to space limitation the details of the proof are provided in [10]. However, we proceed with some

intuition. Consider the decomposing the error as:

$$\hat{H}_1^{(m)} - H(p) = \min_{\theta_1,\ldots,\theta_m} \underbrace{D(p||p_\theta(\lambda^*))}_{\text{Approximation error}} + \underbrace{\sum_{l=1}^{m} \lambda_l^* E_{p-\hat{p}}[\phi_{\theta_l}]}_{\text{Estimation error}} . \tag{3.11}$$

Barron *et. al.* has shown in [8] that ap proximation error can be bounded as $D(p||p_\theta(\lambda^*)) \leq C/m$. Hoeffding's inequality provides a bound on the difference between empirical mean and true mean of a function of *i.i.d.* bounded random variable [61]. By applying the Hoeffding inequality to the estimation error we can bound the estimation error by $\frac{ML}{\sqrt{n}}\sqrt{2(\log\frac{2m}{\delta})}$. To find the rate of convergence based on the number of the samples n, we present the following corollary:

Corollary 7 *Let the number of features used to approximate $p(x)$ be $m = \sqrt{n}$, then with probability $1-\delta$ the estimation error is bounded by*

$$|\tilde{H}_1^{(m)} - H(p)| \leq C_1\sqrt{\frac{\log n}{n}} + o(\sqrt{\frac{\log n}{n}}), \tag{3.12}$$

where $C_1 = CML\sqrt{\frac{1}{2}\log\frac{2}{\delta}}$.

This corollary suggests that the overall error is $\mathcal{O}(\sqrt{\log n/n})$; only a factor of $\sqrt{\log n}$ away from the statistical estimation error $\mathcal{O}(\sqrt{1/n})$. While computationally demanding, the performance of the proposed estimator illustrates the merit in the maximum entropy framework.

3.3.2.2 Greedy m-term entropy estimator

Greedy approaches for approximating functions with m-terms from a given dictionary D were shown to be effective [41]. Greedy m-term approximations offer a computationally efficient alternative to joint optimization of m-term approximations. We consider the greedy approach for the following entropy estimator due to its computational efficiency. We would like to arrive to the m-term approximation of $\log p(x)$, of the form $g_m(x) = \sum_{j=1}^{m} \lambda_j \phi_{\theta_j}(x)$ by adding one term at a time. Start by initializing $g_0(x) = 0$. The lth iteration considers constructing $g_l(x)$ based on $g_{l-1}(x)$ through

$$g_l(x) = (1 - \frac{1}{l}) g_{l-1}(x) + \frac{1}{l} \beta \phi_\theta(x), \tag{3.13}$$

where β and θ are obtained by

$$\min_{\beta,\theta} Z(g_l(x)) - E_{\hat{p}}[g_l(x)]. \tag{3.14}$$

The minimization in (3.14) is convex w.r.t. β when θ is held fixed leaving the main difficulty to optimization w.r.t only a single variable θ. After m iterations, we obtain $g_m(x)$ of the form $g_m(x) = \sum_{j=1}^{m} \lambda_j \phi_{\theta_j}(x)$. Substituting the values of $\{\lambda_j, \phi_{\theta_j}\}_{j=1}^{m}$ into

$$\hat{H}_2^{(m)} = Z(\lambda) - \sum_{j=1}^{m} \lambda_j E_{\hat{p}}[\phi_{\theta_j}], \tag{3.15}$$

yields the proposed entropy estimate. Despite the potential sub-optimality of the greedy approach, the method provides consistent entropy estimates. Its accuracy is examined in the following theorem.

Theorem 8 *For $\hat{H}_2^{(m)}$ defined in (3.15) and $m = \sqrt{n}$ with probability at least $1 - \delta$,*

$$|\hat{H}_2^{(m)} - H(p)| \leq \frac{K_1 1 + \log m}{m} + \frac{K_2}{m} + \frac{K_3}{m^2} \tag{3.16}$$

for $m \geq 4$, where $K_1 \leq 8LM\sqrt{2\log\frac{1}{2\delta}}$, $K_2 \leq 8L^2M^2$, and $K_3 \leq \bar{K_3}$[1].

Due to space limitation, we omit the proof for this theorem, which is available in [10]. Similar to Theorem 6, this bound decomposes the error into approximation error and estimation error. The first term on the RHS is related to the estimation error where the second and third terms are related to the approximation error. We proceed with a corollary, which expresses the convergence rate of the algorithm in terms of the number of samples n.

Corollary 9 *If we select the number of terms m as $m = \sqrt{n}$ in $\tilde{H}_2^{(m)}$ from (3.15), the estimation error of $\tilde{H}_2^{(m)}$ is bounded with probability $1 - \delta$ by*

$$|\hat{H}_2^{(m)} - H(p)| \leq C_3 0.5\frac{\log n}{n} + o(\sqrt{\frac{\log n}{n}}),$$

where $C_3 = K_1 + K_2 + K_3$.

While the greedy method is typically expected to present performance inferior to that of the brute-force estimator, its asymptotic error is of the same order. From a computational point of view, the greedy approach is significantly faster than the brute force method. We proceed with the computationally efficient greedy m-term estimator. In the next section, the performance of the estimator is numerically evaluated and compared to alternatives.

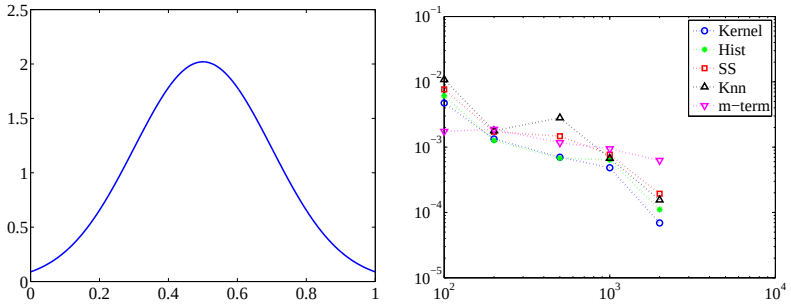
[1] $\bar{K_3} = 48(\frac{32L^9M^9}{81} + \frac{16L^8M^8}{9} + \frac{40L^7M^7}{9} + \frac{20L^6M^6}{3} + \frac{20L^5M^5}{3} + 4L^4M^4 + \frac{8L^3M^3}{3})$

3.4 Simulations

In this part we compare the performance of well known entropy estimation approaches with the greedy m-term estimator defined in Section 3.3.2.2 on data drawn from three univariate continuous distributions as well as on experimental sensor network data. The estimators considered in this comparison study are: (i) *Histogram*: the plug-in estimator for the histogram density estimation using a constant bins width chosen according to [101]. (ii) *KDE*: the kernel density estimator with the optimal bandwidth chosen according to [24]. (iii) *Sample spacing*: the classical sample spacing approach with $m = 5$. (iv) *Nearest neighbors*: the nearest neighbor estimator with $k = 5$. (v) *Greedy m-term*: the proposed approach with a dictionary of 1500 features ϕ including polynomials x^i and trigonometric basis $[\sin(2\pi ix), \cos(2\pi ix)]$ for $i = 1, \ldots, 500$. To indicate the number of terms m, we used the L_1 norm to restrict the complexity of the approximated function.

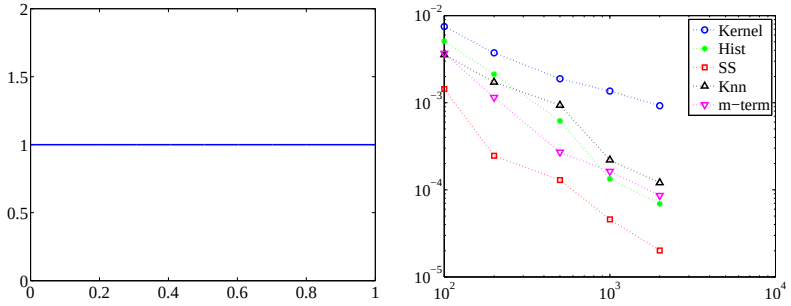
3.4.1 Synthetic dataset

We consider three univariate distributions: *truncated normal* with $\mu = 0.5$ and $\sigma = 0.2$, *uniform* between $(0, 1)$, and truncated mixture of five Gaussians with $\mu = [0.3, 0.5, 0.7, 0.8, 0.85]$ and $\sigma = [0.09, 0.01, 0.009, 0.001, 0.0005]$ respectively. For each distribution, samples of size $[100, 200, 500, 1000, 2000]$ were considered and 10 runs of the experiment were conducted. The left column of Fig. 3.4 depicts the distributions and the right column shows the accuracy of algorithms in terms of mean square error. For the two simple classical example (*truncated normal*, *uniform*) all algorithms perform very closely. However, in the mixture of Gaussians example m-term estimator outperforms the other algorithms. Note that there is no Gaussian basis in the dictionary D.



(a) (b)

Figure 3.2: Toy examples



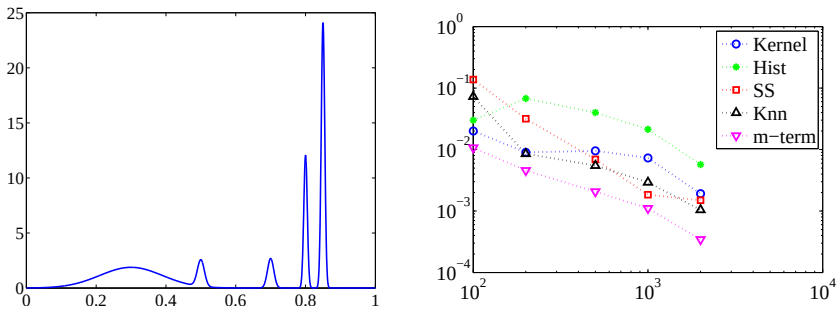
(a) (b)

Figure 3.3: Toy examples

The approximation of five mixture of Gaussians using the m-term approximation was performed and the result is depicted in Fig. 3.5. This example illustrates the approximation of the true density. The figure is in log scale since $\log p(x)$ is approximated by a linear combination of the features ϕ_j's.

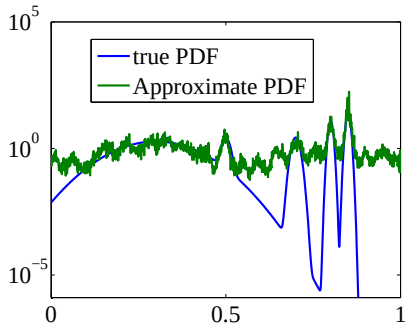
3.4.2 Anomaly detection in sensor network

We considered the use of the greedy m-term estimator for anomaly detection. An experiment was set up on a Mica2 platform, which consists of 14 sensor nodes randomly deployed inside and outside a lab room. Wireless sensors communicate with each other



(a) (b)

Figure 3.4: Toy examples

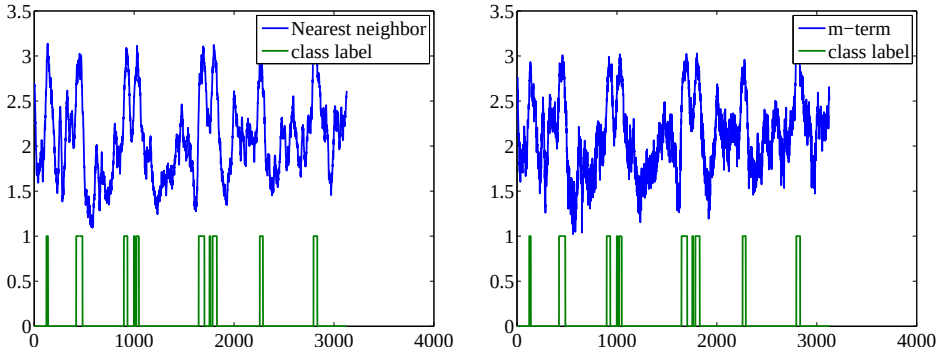


(a)

Figure 3.5: Graph of $p(x)$ vs. the approximated $p(x)$ using m-term approximation approach

by broadcasting and the received signal strength (RSS), defined as the voltage measured by a receiver's received signal strength indicator circuit (RSSI), was recorded for each pair of transmitting and receiving nodes. There were $14 \times 13 = 182$ pairs of RSSI measurements over a 30 minute period, and each sample was acquired every 0.5 sec. During the measuring period, students walked into and out of lab at random times, which caused anomaly patterns in the RSSI measurements. Finally, a web camera was employed to record activity for ground truth. The mission of this experiment is to use the 182 RSS sequences to detect any intruders (anomalies). Fig. 3.6 shows the results of the greedy m-term estimator and *nearest neighbor*. Due to space limitation, we omitted

the results of other algorithms on this dataset. We observe that the entropy peaks at times of anomaly in a similar fashion for both methods. Though the two methods are based on different frameworks, similar entropy estimates are produced.



(a) (b)

Figure 3.6: Anomaly detection in sensor network data using the nearest neighbor and m-term estimator

Chapter 4: Convergence Analysis for Entropy Estimation Using the Principle of Maximum Entropy

4.1 Introduction

In this report we analyze the error of entropy estimation for an unknown density function $p(x)$ using the principle of maximum entropy approach. We propose two estimators for entropy estimation which is called brute-force and greedy m-term approximation. The derivation of the error bound of two estimators is provided here. First, we start with the definition of the problem, model assumptions, and restrictions. Then we define the estimators and prove the bound on the error for each estimator.

4.2 problem definition

We are given n i.i.d. samples $x_1, \ldots, x_n$ from an unknown probability density function $p(x)$ and we want to estimate the entropy which is defined by

$$H(p) = -E_p[\log p(x)] = -\int_{\mathcal{X}} p(x) \log p(x) dx. \tag{4.1}$$

where $\mathcal{X}$ is a bounded support. We use the principle of maximum entropy to approximate $p(x)$ and then use the definition (4.1) to estimate the entropy.

4.2.1 Principle of maximum entropy

For a set of m feature functions $\phi_{\theta_l}(x)$'s ($l = 1, \ldots, m$) over the space of the data samples $\mathcal{X}$, maximum entropy framework among all density functions fits a density function which is consistent with the set of constraints and otherwise is uniform as follows:

$$\begin{aligned} &\max_{p(x)} H(p) \\ &\quad \text{s.t.} \\ &\quad E_p[\phi_{\theta_l}(x)] = E_{\hat{p}}[\phi_{\theta_l}(x)], \end{aligned} \tag{4.2}$$

where $E_{\hat{p}}[g(x)] = \frac{1}{n}\sum_{i=1}^{n} g(x_i)$ is the empirical mean of $g(x)$. The obtained solution from (4.2) is a general form of distribution in the class of exponential family which can be represented as:

$$p(x; \underline{\lambda}) = e^{\sum_{l=1}^{m} \lambda_l \phi_l(x) - Z(\underline{\lambda})}, \tag{4.3}$$

where $\underline{\lambda} = [\lambda_1, \ldots, \lambda_m]$ are the Lagrangian multipliers correspond to the set of constraints $E_p[\phi_l(x)] = E_{\hat{p}}[\phi_l(x)]$, and $Z(\underline{\lambda}, \underline{\theta}) = \log \int e^{\sum_{l=1}^{m} \lambda_l \phi_l(x)}$. Substituting the PDF given by (4.3) into (4.1), yields a parametric expression for the entropy:

$$H(p(x; \underline{\lambda})) = Z(\underline{\lambda}) - \sum_{l=1}^{m} \lambda_l E_{p(x;\underline{\lambda})}[\phi_l(x)]. \tag{4.4}$$

The set of PDFs $\mathcal{P} = \{p(x; \underline{\lambda}) | \underline{\lambda} \in \Lambda\}$ provides an approximation space for $p(x)$. We propose the following estimator:

$$\hat{H}^{(m)} = Z(\hat{\underline{\lambda}}) - \sum_{l=1}^{m} \hat{\lambda}_l E_{\hat{p}}[\phi_l(x)], \tag{4.5}$$

where

$$\hat{\underline{\lambda}} = \arg\min_{\underline{\lambda}} Z(\underline{\lambda}) - \sum_{l=1}^{m} \lambda_l E_{\hat{p}}[\phi_l(x)]. \tag{4.6}$$

$p(x;\underline{\lambda})$ in (3.4m) is the one which maximizes the entropy for the set of given $\phi_l(x)$. To minimize the entropy, we search over the space of the feature functions $\phi_{\theta_l}(x)$, $\theta \in \Theta$ to find the best set of the feature functions which minimizes the entropy. Thus, the estimator is defined as:

$$\hat{H}^{(m)} = \min_{\underline{\theta},\underline{\lambda}} Z(\underline{\lambda},\underline{\theta}) - \sum_{l=1}^{m} \lambda_l E_{\hat{p}}[\phi_{\theta_l}(x)]. \tag{4.7}$$

4.3 Approximation and model assumption

Suppose we have a continuous function $f \in C^1$, defined on a compact interval $f : C[\mathcal{R}^d] \to \mathcal{R}$. Based on Weierstrass-Stone theorem every continuous function on the compact interval can be approximated uniformly by polynomials. It means the polynomial functions on the compact interval are dense enough to approximate any continuous function on that interval. Note that the only requirement for Weierstrass-Stone theorem is the continuity of f on the compact interval.

For example for a given polynomial basis $\phi_\theta(x) = \{x^\theta | \theta \in \mathcal{N}\}$, we can write function f as a linear expansion of the basis ϕ_θ as follows:

$$f = \int \phi_\theta(x)\lambda(\theta)\mu(d\theta) \tag{4.8}$$

where $\lambda(\theta)$ is the coefficient corresponding to the basis function $\phi_\theta(x)$ and μ is the measurement defined on the space of Θ. We use the same idea to approximate the

$\log p(x)$ on the compact interval. We assume that $\log p(x)$ is continuous on the compact interval and it can be written as a linear expansion of basis $\phi_\theta(x)$ as follows:

$$\log p(x) = \int \phi_\theta(x)\lambda(\theta)\mu(d\theta), \tag{4.9}$$

where $\phi_\theta(x) \in \Phi_1$, and $\lambda(\theta) \in \mathcal{R}$. Note that the set of Φ_1 is not restricted to the polynomial basis and it includes all the basis functions such as trigonometric, and splines basis functions. However, the expansion in (4.9) always exists for $\log p(x)$ based on the Weierstrass-Stone theorem by just having the polynomials as feature functions. The continuity of $\log p(x)$ implies that $\| \log p(x)\|_\infty$ is finite.

To make sure that $\| \log p(x)\|_\infty$ is finite, we make the assumptions that $\|\phi_\theta\|_\infty \leq M$, and $\int |\lambda(\theta)|\mu(d\theta) \leq L$ are finite. Because $\lambda(\theta) \in \mathcal{R}$, it can be positive or negative. Moreover we assume $p(x)$ has bounded support (e.g. $\int_{\mathcal{X}} dx = C$). To make $\lambda(\theta)$ always positive, we define $\Phi_2 = \{-\phi_\theta(x)|\theta \in \Theta\}$ and $\Phi = \Phi_1 \cup \Phi_2$. Thus, we redefine $\log p$ as follows:

$$\log p(x) = \int \phi_\theta(x)\tilde{\lambda}(\theta)\mu(d\theta) \tag{4.10}$$

where $\phi_\theta(x) \in \Phi$ and $\tilde{\lambda}(\theta) \geq 0$. Having $\tilde{\lambda}(\theta) \geq 0$, we can define a probability measure on the space of Θ by $\tilde{\lambda}(\theta)$.

4.4 Entropy estimators

We propose a brute force m-term entropy estimator, and a greedy m-term estimator to estimate the entropy based on the maximum entropy framework. In the brute force approach we optimize the estimator w.r.t. the parameters λ and θ jointly, where in

the greedy approach the optimization is done in m step and in each step we optimize the estimator for one value of θ and the coefficient corresponds to it. The brute force optimization is a challenging task due to the presence of parameter θ which makes the optimization non-convex. Moreover, there is no straight forward way of optimizing the estimator over the space of Θ jointly. On the other hand, the greedy approach provides a convenient way of handling this problem. In the following each estimator is explained and a bound for the error of estimation in each case is proposed. To be able to bound the error we need to make some restrictions on the space of the parameters of the estimators which is explained in detail in each section.

4.4.1 Brute-force m-term entropy estimator

Problem definition

We propose the following estimator

$$\hat{H}_1^{(m)} = \min_{\underline{\theta},\|\underline{\lambda}\|_1\leq L} Z(\underline{\theta},\underline{\lambda}) - \sum_{l=1}^{m} \lambda_l E_{\hat{p}}\phi_{\theta_l}(x). \tag{4.11}$$

To be able to bound the error of estimation, we restrict $\|\underline{\lambda}\|_1 \leq L$. The feature functions $\phi_\theta(x)$ is also bounded $\|\phi_\theta(x)\|_\infty \leq M$. We define $\hat{\underline{\lambda}}(\theta)$ and $\underline{\lambda}^*(\theta)$ as follows:

$$\hat{\underline{\lambda}}(\theta) = \arg\min_{\|\lambda\|_1\leq L} Z(\underline{\theta},\underline{\lambda}) - \sum_{l=1}^{m} \lambda_l E_{\hat{p}}\phi_{\theta_l}(x) \tag{4.12}$$

$$\underline{\lambda}^*(\theta) = \arg\min_{\|\lambda\|_1\leq L} Z(\underline{\theta},\underline{\lambda}) - \sum_{l=1}^{m} \lambda_l E_{p}\phi_{\theta_l}(x) \tag{4.13}$$

Theorem 10 *$\forall\underline{\lambda}, \|\underline{\lambda}\| \leq L$, and $\|\phi_\theta\|_\infty \leq M$ the estimation error associated with $\hat{H}_1^{(m)}$*

with probability at least $1-\delta$ *satisfies:*

$$-\frac{ML\sqrt{d+2\log m}}{\sqrt{n}} \leq \hat{H}_1^{(m)} - H(p) \leq \frac{C_1}{2m} + \frac{ML\sqrt{d+2\log m}}{\sqrt{n}} \tag{4.14}$$

where $d = 2\log\frac{2}{\delta}$*, and* $C_1 = 27(e^{2LM} - 1 - 2LM - 14/9L^2M^2)$.

We define the error of estimation in this approach $E^{(m)}$ as follows:

$$E^{(m)} = \hat{H}_1^{(m)} - H(p) \quad = \quad \min_{\underline{\theta}, \|\underline{\lambda}\| \leq L} Z(\underline{\lambda}; \underline{\theta}) - \sum_{l=1}^{m} \lambda_l E_{\hat{p}}[\phi_{\theta_l}(x)] - H(p), \tag{4.15}$$

and separately obtain RHS and LHS inequality in (4.14).

4.4.1.1 Right hand side inequality

In this part we want to show that $E^{(m)}$ is bounded above by

$$E^{(m)} \leq \frac{C_1}{m} + \frac{ML\sqrt{d+2\log m}}{\sqrt{n}}, \tag{4.16}$$

where $d = 2\log\frac{2}{\delta}$.

Proof *We start with* $E^{(m)}$ *as defined in (4.15) as follows:*

$$\begin{aligned} E^{(m)} &= \min_{\underline{\theta}, \|\underline{\lambda}\|_1 \leq L} Z(\underline{\lambda}; \underline{\theta}) - \sum_{l=1}^{m} \lambda_l E_{\hat{p}}[\phi_{\theta_l}(x)] - H(p) \\ &= \min_{\underline{\theta}} \min_{\|\underline{\lambda}\|_1 \leq L} Z(\underline{\lambda}; \underline{\theta}) - \sum_{l=1}^{m} \lambda_l E_{\hat{p}}[\phi_{\theta_l}(x)] - H(p) \\ &= \min_{\underline{\theta}} Z(\hat{\underline{\lambda}}(\theta); \underline{\theta}) - \sum_{l=1}^{m} \hat{\lambda}_l(\theta) E_{\hat{p}}[\phi_{\theta_l}(x)] - H(p) \end{aligned} \tag{4.17}$$

Since $\hat{\lambda}_l(\theta)$ is the minimizer of (4.17), for $\underline{\lambda}^*(\theta)$, we have:

$$\begin{aligned} E^{(m)} &\leq \min_{\underline{\theta}} Z(\underline{\lambda}^*(\theta);\underline{\theta}) - \sum_{l=1}^{m} \lambda_l^*(\theta) E_{\hat{p}}[\phi_{\theta_l}(x)] - H(p) \\ &= \min_{\underline{\theta}} Z(\underline{\lambda}^*(\theta);\underline{\theta}) - \sum_{l=1}^{m} \lambda_l^*(\theta) E_p[\phi_{\theta_l}(x)] + \sum_{l=1}^{m} \lambda_l^*(\theta)(E_p[\phi_{\theta_l}(x)] - E_{\hat{p}}[\phi_{\theta_l}(x)]) - H(p) \\ &= \min_{\underline{\theta}} D(p||p(x;\underline{\lambda}^*(\theta),\underline{\theta})) + \sum_{l=1}^{m} \lambda_l^*(\theta)(E_p[\phi_{\theta_l}(x)] - E_{\hat{p}}[\phi_{\theta_l}(x)]), \end{aligned} \tag{4.18}$$

where $D(p||p(x;\underline{\lambda}^*(\theta),\underline{\theta})) = E_p[\log p - \log p(x;\underline{\lambda}^*(\theta),\underline{\theta})]$. Note that the RHS of (4.18) is obtained by adding and subtracting $\sum_{l=1}^{m} \lambda_l^*(\theta) E_p[\phi_{\theta_l}(x)]$. The first term in RHS of (4.18) is the approximation error and the second term is the estimation error. The estimation error can be bounded by applying Hoeffding inequality with probability at least $1-\delta$ as follows (see Appendix 4.5.2):

$$|\sum_{l=1}^{m} \lambda_l^*(\theta)(E_p[\phi_{\theta_l}(x)] - E_{\hat{p}}[\phi_{\theta_l}(x)])| \leq \frac{ML\sqrt{d+2\log m}}{\sqrt{n}} \tag{4.19}$$

Plugging back (4.19) into (4.18) yields:

$$E^{(m)} \leq \frac{ML\sqrt{d+2\log m}}{\sqrt{n}} + \min_{\underline{\theta}} D(p||p(x;\underline{\lambda}^*(\theta),\underline{\theta})) \tag{4.20}$$

$\min_{\underline{\theta}} D(p||p(x;\underline{\lambda}^*(\theta),\underline{\theta}))$ can be bounded by $\min_{0\leq\alpha_l\leq 1,|\beta|\leq L,\theta_l} D(p||p_{g_l})$, where p_{g_l} is obtained by a greedy approach. $\min_{0\leq\alpha_l\leq 1,|\beta|\leq L,\theta_l} D(p||p_{g_l})$ can be bounded as follows (see Appendix 4.5.1):

$$\begin{aligned} \min_{0\leq\alpha_l\leq 1,|\beta|\leq L,\theta_l} D(p||p_{g_l}) &\leq \frac{12L^2M^2}{l+2} + \frac{27(e^{2LM} - 1 - 2LM - 2L^2M^2)}{(l+2)^2} \\ &\leq \frac{12L^2M^2 + 27(e^{2LM} - 1 - 2LM - 2L^2M^2)}{(l+2)} \end{aligned}$$

$$\leq \quad \frac{12L^2M^2 + 27(e^{2LM} - 1 - 2LM - 2L^2M^2)}{l}. \quad (4.21)$$

If we put $l = m$ in (4.21) for the error in step m, thus

$$\min_{\underline{\theta}} D(p||p(x; \underline{\lambda}^*(\theta), \underline{\theta})) \quad \leq \quad \frac{C_1}{m}, \quad (4.22)$$

where $C_1 = 27(e^{2LM} - 1 - 2LM - 14/9L^2M^2)$. Plugging back (4.22) into (4.20) yields (4.16). ■

4.4.1.2 Left hand side inequality

We want to show that:

$$E^{(m)} \geq -\frac{ML\sqrt{d + 2\log m}}{\sqrt{n}}, \quad (4.23)$$

where $d = 2\log\frac{2}{\delta}$.

Proof *We start with*

$$E^{(m)} = \min_{\underline{\theta}} Z(\hat{\underline{\lambda}}(\theta); \underline{\theta}) - \sum_{l=1}^{m} \hat{\lambda}_l(\theta) E_{\hat{p}}[\phi_{\theta_l}(x)] - H(p), \quad (4.24)$$

and reorganize it as follows:

$$\begin{aligned} E^{(m)} &= \min_{\underline{\theta}} Z(\hat{\underline{\lambda}}(\theta); \underline{\theta}) - \sum_{l=1}^{m} \hat{\lambda}_l(\theta) E_p[\phi_{\theta_l}(x)] - \sum_{l=1}^{m} \hat{\lambda}_l(\theta)(E_{\hat{p}}[\phi_{\theta_l}(x)] - E_p[\phi_{\theta_l}(x)]) - H(p) \\ &= \min_{\underline{\theta}} D(p||p(x; \hat{\underline{\lambda}}(\theta), \underline{\theta})) - \sum_{l=1}^{m} \hat{\lambda}_l(\theta)(E_{\hat{p}}[\phi_{\theta_l}(x)] - E_p[\phi_{\theta_l}(x)]). \quad (4.25) \end{aligned}$$

Since $\min_{\underline{\theta}} D(p||p(x;\hat{\underline{\lambda}}(\theta),\underline{\theta})) \geq 0$, *therefore,*

$$E^{(m)} \geq -\sum_{l=1}^{m} \hat{\lambda}_l(\theta)(E_{\hat{p}}[\phi_{\theta_l}(x)] - E_p[\phi_{\theta_l}(x)]). \tag{4.26}$$

Applying Hoeffding inequality to the RHS of (4.26) with probability at least $1-\delta$, *we have*

$$-|\sum_{l=1}^{m} \hat{\lambda}_l(\theta)(E_p[\phi_{\theta_l}(x)] - E_{\hat{p}}[\phi_{\theta_l}(x)])| \geq -\frac{ML\sqrt{d+2\log m}}{\sqrt{n}} \tag{4.27}$$

(see Appendix 4.5.2). Plugging back (4.27) into (4.26) yields (4.23). ■

4.4.2 Greedy m-term approximation

We consider approxiamting $\log p(x)$ by $g_l - Z(g_l)$ where $g_l = \sum_{k=1}^{l} \lambda_l \phi_{\theta_l}(x)$ as in the previous section and $Z(g_l) = \int e^{g_l} dx$. Let $g_0 = 0$. We construct g_l recursively according to $g_l^* = g_l(\alpha_l^*, \beta_l^*, \theta_l^*)$, where

$$g_l(\alpha_l, \beta_l, \theta_l) = (1-\alpha_l)g_{l-1} + \alpha_l \beta_l \phi_{\theta_l} \quad l = 1, \ldots, m. \tag{4.28}$$

and

$$\alpha_l^*, \beta_l^*, \theta_l^* = \arg \min_{0 \leq \alpha_l \leq 1, |\beta_l| \leq L, \theta_l} Z(g_l(\alpha_l, \beta_l, \theta_l)) - E_{\hat{p}}[g_l(\alpha_l, \beta_l, \theta_l)]. \tag{4.29}$$

Note that to be able to bound the error we restrict $0 \leq \alpha_l \leq 1$, $|\beta_l| \leq L$, and $\|\theta_l\|_\infty \leq M$. We propose the following approximation:

$$\hat{H}_2^{(l)} \quad = \quad Z(g_l^*) - E_{\hat{p}}[g_l^*],\ l = 1, \ldots, m. \tag{4.30}$$

We define the error associated with $\hat{H}_2^{(l)}$ as follows:

$$E^{(l)} = \hat{H}_2^{(l)} - H(p) \quad = \quad \min_{0 \leq \alpha_l \leq 1, |\beta_l| \leq L, \theta_l} Z(g_l) - E_{\hat{p}}[g_l] - H(p). \tag{4.31}$$

Theorem 11 *For $m = \sqrt{n}$, and $\|\phi_{\theta_l}\|_\infty \leq M$, the estimation error associated with $\hat{H}_2^{(m)}$ with probability at least $1 - \delta$ satisfies:*

$$-\frac{K_1\sqrt{d + 2\log m}}{m} \leq \hat{H}_2^{(m)} - H(p) \leq \frac{K_1\sqrt{d + 2\log m}}{m} + \frac{6K_2}{m+2} + \frac{27K_3}{(m+2)^2}, \tag{4.32}$$

where $d = 2\log\frac{2}{\delta}$, $K_1 = 2LM$, $K_2 = 2L^2M^2$, and $K_3 = e^{2LM} - 1 - 2LM - 2L^2M^2$.

We start with the definition of the error in step l and express it in terms of the error in step $l-1$. This recursion helps to configure how the error decays in each step by adding one term at a time.

4.4.2.1 Right hand side inequality

We want to show that with probability at least $1 - \delta$

$$E^{(m)} \leq \frac{K_1\sqrt{d + 2\log m}}{m} + \frac{6K_2}{m+2} + \frac{27K_3}{(m+2)^2}. \tag{4.33}$$

Error recursion In this part we want to show that with probability at least $1-\delta$, the recursion error is as follows:

$$E^{(l)} \leq \min_{0\leq\alpha_l\leq 1}(1-\alpha_l)E^{(l-1)} + \frac{K_1\alpha_l\sqrt{d+2\log m}}{\sqrt{n}} + \alpha_l^2 K_2 + \alpha_l^3 K_3, \quad (4.34)$$

where $d = 2\log\frac{2}{\delta}$, $K_1 = LM$, $K_2 = 2L^2M^2$,$K_3 = e^{2LM}-1-2LM-2L^2M^2$, $\|\phi_{\theta_l}\|_\infty \leq M$, and $|\beta_l| \leq L$.

Equality error recursion To obtain (4.34), we relate error in step l to the error in step $l-1$ in terms of equality. In other word, we are looking for a relation as follows:

$$E^{(l)} = \min_{0\leq\alpha_l\leq 1}(1-\alpha_l)E^{(l-1)} + G(\alpha_l), \quad (4.35)$$

where $G(\alpha_l)$ is

$$G(\alpha_l) = \min_{|\beta_l|\leq L,\theta_l}(D(p||p_{g_l}) - D(p||p_{g_{l-1}})) + \alpha_l(D(p||p_{g_{l-1}}) + E_p[\beta_l\phi_{\theta_l}] - E_{\hat{p}}[\beta_l\phi_{\theta_l}]) \quad (4.36)$$

Proof *We start with (4.31) and add and subtract term $E_p[g_l]$, to express the RHS in terms of the approximation error and estimation error as follows:*

$$\begin{aligned} E^{(l)} &= \min_{0\leq\alpha_l\leq 1,|\beta_l|\leq L,\theta_l} Z(g_l) - E_{\hat{p}}[g_l] - H(p) \\ &= \min_{0\leq\alpha_l\leq 1,|\beta_l|\leq L,\theta_l} Z(g_l) - E_p[g_l] - H(p) + E_p[g_l] - E_{\hat{p}}[g_l] \\ &= \min_{0\leq\alpha_l\leq 1,|\beta_l|\leq L,\theta_l} D(p||p_{g_l}) + (E_p[g_l] - E_{\hat{p}}[g_l]). \end{aligned} \quad (4.37)$$

Plugging back g_l from (4.28) into the second term in the RHS of (4.37) yields:

$$\begin{aligned} E_p[g_l] - E_{\hat{p}}[g_l] &= E_p[(1-\alpha_l)g_{l-1} + \alpha_l\beta_l\phi_{\theta_l}] - E_{\hat{p}}[(1-\alpha_l)g_{l-1} + \alpha_l\beta_l\phi_{\theta_l}] \\ &= (1-\alpha_l)(E_p[g_{l-1}] - E_{\hat{p}}[g_{l-1}]) + \alpha_l(E_p[\beta_l\phi_{\theta_l}] - E_{\hat{p}}[\beta_l\phi_{\theta_l}]). \end{aligned} \quad (4.38)$$

Substitute (4.38) into (4.37) yields

$$\begin{aligned} E^{(l)} = &\min_{0\le\alpha_l\le 1,|\beta_l|\le L,\theta_l} D(p||p_{g_l}) + (1-\alpha_l)(E_p[g_{l-1}] - E_{\hat{p}}[g_{l-1}]) + \\ &\alpha_l(E_p[\beta_l\phi_{\theta_l}] - E_{\hat{p}}[\beta_l\phi_{\theta_l}]). \end{aligned} \quad (4.39)$$

If we add and subtract $(1-\alpha_l)D(p||p_{g_{l-1}})$ to (4.39) yields

$$\begin{aligned} E^{(l)} = &\min_{0\le\alpha_l\le 1,|\beta_l|\le L,\theta_l} (1-\alpha_l)(D(p||p_{g_{l-1}}) + E_p[g_{l-1}] - E_{\hat{p}}[g_{l-1}]) + D(p||p_{g_l}) + \\ &\alpha(E_p[\beta_l\phi_{\theta_l}] - E_{\hat{p}}[\beta_l\phi_{\theta_l}]) - (1-\alpha_l)D(p||p_{g_{l-1}}) \\ = &\min_{0\le\alpha_l\le 1,|\beta_l|\le L,\theta_l} (1-\alpha_l)E^{(l-1)} + (D(p||p_{g_l}) - D(p||p_{g_{l-1}})) \\ &+\alpha(D(p||p_{g_{l-1}}) + E_p[\beta_l\phi_{\theta_l}] - E_{\hat{p}}[\beta_l\phi_{\theta_l}]), \end{aligned} \quad (4.40)$$

where $E^{(l-1)} = D(p||p_{g_{l-1}}) + E_p[g_{l-1}] - E_{\hat{p}}[g_{l-1}]$. ■

Inequality error recursion By bounding $G(\alpha_l) \le F(\alpha_l)$, we show that with probability at least $1-\delta$

$$E^{(l)} \le \min_{0\le\alpha_l\le 1} (1-\alpha_l)E^{(l-1)} + F(\alpha_l), \quad (4.41)$$

where $F(\alpha_l)$ is

$$F(\alpha_l) = \alpha_l K_1' + \alpha_l^2 K_2 + \alpha_l^3 K_3, \tag{4.42}$$

and $K_1' = \frac{\sqrt{d+2\log m}}{\sqrt{n}} K_1$, $K_1 = LM$, $K_2 = 2L^2M^2$, $K_3 = e^{2LM} - 1 - 2LM - 2L^2M^2$.

Proof *We start with $G(\alpha_l)$ as follows:*

$$\begin{aligned} G(\alpha_l) &= \min_{|\beta_l| \le L, \theta_l} (D(p||p_{g_l}) - D(p||p_{g_{l-1}})) + \alpha_l (D(p||p_{g_{l-1}}) + E_p[\beta_l \phi_{\theta_l}] - E_{\hat{p}}[\beta_l \phi_{\theta_l}]) \\ &= \alpha_l D(p||p_{g_{l-1}}) + \min_{|\beta_l| \le L, \theta_l} (D(p||p_{g_l}) - D(p||p_{g_{l-1}})) \\ &+ \alpha_l (E_p[\beta_l \phi_{\theta_l}] - E_{\hat{p}}[\beta_l \phi_{\theta_l}]). \end{aligned} \tag{4.43}$$

The second term on the RHS of (4.43) is a random quantity and can be bounded using the Hoeffding inequality with probability at least $1 - \delta$ (see Appendix 4.5.2) as follows:

$$|E_p[\beta_l \phi_{\theta_l}] - E_{\hat{p}}[\beta_l \phi_{\theta_l}]| \le \frac{\sqrt{d + 2\log m}}{\sqrt{n}} K_1 = K_1', \tag{4.44}$$

where $d = 2\log\frac{2}{\delta}$, and $K_1 = LM$. Plugging back (4.44) into (4.43), with probability at least $1 - \delta$

$$G(\alpha_l) \;\; \le \;\; \alpha_l (D(p||p_{g_{l-1}}) + K_1') + \min_{|\beta_l| \le L, \theta_l} (D(p||p_{g_l}) - D(p||p_{g_{l-1}})), \tag{4.45}$$

where $K_1' = \frac{\sqrt{d+2\log m}}{\sqrt{n}} K_1$. We bound term $\min_{|\beta_l| \le L, \theta_l} (D(p||p_{g_l}) - D(p||p_{g_{l-1}}))$ as follows:

$$\begin{aligned} \min_{|\beta_l| \le L, \theta_l} D(p||p_{g_l}) - D(p||p_{g_{l-1}}) &= \min_{|\beta_l| \le L, \theta_l} E_p[\log p] - E_p[\log p_{g_l}] - E_p[\log p] + E_p[\log p_{g_{l-1}}] \\ &= \min_{|\beta_l| \le L, \theta_l} E_p[g_{l-1} - Z(g_{l-1})] - E_p[g_l - Z(g_l)] \end{aligned}$$

$$
\begin{aligned}
&= \min_{|\beta_l|\leq L,\theta_l} E_p[g_{l-1} - g_l] + E_p[Z(g_l) - Z(g_{l-1})] \\
&= \min_{|\beta_l|\leq L,\theta_l} -\alpha_l E_p[\Delta] + E_p[\log \frac{\int e^{g_{l-1}} e^{\alpha_l \Delta_l}}{\int e^{g_{l-1}}}], \\
&= \min_{|\beta_l|\leq L,\theta_l} -\alpha_l E_p[\Delta_l] + E_p[\log(E_{p_{g_{l-1}}}[e^{\alpha_l \Delta_l}])] \qquad (4.46)
\end{aligned}
$$

where $\Delta_l = \beta_l \phi_{\theta_l} - g_{l-1}$. *To bound the log term, we use the inequality* $\log(1+\epsilon) \leq \epsilon$ *and set* $\epsilon = E_{p_{g_{l-1}}}[e^{\alpha_l \Delta_l}] - 1$. *Thus,*

$$
\min_{|\beta_l|\leq L,\theta_l} D(p||p_{g_l}) - D(p||p_{g_{l-1}}) \leq \min_{|\beta_l|\leq L,\theta_l} -\alpha_l E_p[\Delta_l] + (E_{p_{g_{l-1}}}[e^{\alpha_l \Delta_l}] - 1) \qquad (4.47)
$$

To bound $e^{\alpha_l \Delta_l}$ *for* $||\Delta_l||_\infty \leq 2LM$, *we use Taylor series expansion as follows:*

$$
e^{\alpha_l \Delta_l} \leq 1 + \alpha_l \Delta_l + \frac{\alpha_l^2 \Delta_l^2}{2} + C_3 \frac{\alpha_l^3 |\Delta_l^3|}{6}, \qquad (4.48)
$$

where $C_3 = \frac{e^{2LM} - 1 - 2LM - \frac{4L^2M^2}{2}}{\frac{8L^3M^3}{6}}$. *Note that* g_l, *and* Δ_l *can be bounded as follows:*

$$
||g_l||_\infty \leq |(1-\alpha_l)|||g_{l-1}||_\infty + |\alpha_l||\beta_l|||\phi_{\theta_l}||_\infty. \qquad (4.49)
$$

Since $|\beta_l| \leq L$, $||\phi_{\theta_l}||_\infty \leq M$, $||g_{l-1}||_\infty \leq LM$, *and* $0 \leq \alpha_l \leq 1$ *therefore*

$$
\begin{aligned}
||g_l||_\infty &\leq (1-\alpha_l)LM + \alpha_l LM \\
&\leq LM, \qquad (4.50)
\end{aligned}
$$

and

$$
\begin{aligned}
\Delta_l &\leq ||\Delta_l||_\infty \\
&= ||\beta_l \phi_{\theta_l} - g_{l-1}||_\infty
\end{aligned}
$$

$$\begin{aligned} &\leq ||\beta_l \phi_{\theta_l}||_\infty + ||g_{l-1}||_\infty \\ &\leq LM + LM = 2LM. \end{aligned} \tag{4.51}$$

Thus,

$$\begin{aligned} \min_{|\beta_l| \leq L, \theta_l} D(p||p_{g_l}) - D(p||p_{g_{l-1}}) &\leq \min_{|\beta_l| \leq L, \theta_l} -\alpha_l E_p[\Delta_l] + \alpha_l E_{p_{g_{l-1}}}[\Delta_l] + \alpha_l^2 E_{p_{g_{l-1}}}[\frac{\Delta_l^2}{2}] \\ &+ \alpha_l^3 E_{p_{g_{l-1}}}[\frac{C_3 |\Delta_l^3|}{6}]. \end{aligned}$$

Using (4.51) we can bound $E_{p_{g_{l-1}}}[\frac{\Delta_l^2}{2}] \leq 2L^2M^2$, *and* $E_{p_{g_{l-1}}}[\frac{C_3|\Delta_l^3|}{6}] \leq e^{2LM} - 1 - 2LM - 2L^2M^2$. *Thus,*

$$\begin{aligned} \min_{|\beta_l| \leq L, \theta_l} D(p||p_{g_l}) - D(p||p_{g_{l-1}}) &\leq \alpha_l^2 K_2 + \alpha_l^3 K_3 + \min_{|\beta_l| \leq L, \theta_l} -\alpha_l E_p[\Delta_l] \\ &+ \alpha_l E_{p_{g_{l-1}}}[\Delta_l], \end{aligned} \tag{4.52}$$

where $K_2 = 2L^2M^2$, *and* $K_3 = e^{2LM} - 1 - 2LM - 2L^2M^2$. *Plugging back (4.52) into (4.45) with probability at least* $1 - \delta$

$$\begin{aligned} G(\alpha_l) &\leq \alpha_l (D(p||p_{g_{l-1}}) + K_1') + \alpha_l^2 K_2 + \alpha_l^3 K_3 \\ &+ \min_{|\beta_l| \leq L, \theta_l} \alpha_l (E_{p_{g_{l-1}}}[\Delta_l] - E_p[\Delta_l]). \end{aligned} \tag{4.53}$$

To bound $\min_{|\beta_l| \leq L, \theta_l} \alpha_l (E_{p_{g_{l-1}}}[\Delta_l] - E_p[\Delta_l])$, *we start with simplifying the term*

$$\begin{aligned} \min_{|\beta_l| \leq L, \theta_l} \alpha_l (E_{p_{g_{l-1}}}[\Delta_l] - E_p[\Delta_l]) &= \min_{|\beta_l| \leq L, \theta_l} \alpha_l (E_{p_{g_{l-1}}}[\beta_l \phi_{\theta_l} - g_{l-1}] - E_p[\beta_l \phi_{\theta_l} - g_{l-1}]) \\ &= \alpha_l (E_p[g_{l-1}] - E_{p_{g_{l-1}}}[g_{l-1}]) \\ &+ \min_{|\beta_l| \leq L, \theta_l} \alpha_l (E_{p_{g_{l-1}}}[\beta_l \phi_{\theta_l}] - E_p[\beta_l \phi_{\theta_l}]) \end{aligned}$$

(4.54)

We can simplify $\alpha_l(E_p[g_{l-1}] - E_{p_{g_{l-1}}}[g_{l-1}])$ *as follows:*

$$\begin{aligned}
\alpha_l(E_p[g_{l-1}] - E_{p_{g_{l-1}}}[g_{l-1}]) &= \alpha_l(E_p[g_{l-1} - Z(g_{l-1})] + Z(g_{l-1}) \\
&- E_{p_{g_{l-1}}}[g_{l-1} - Z(g_{l-1})] - Z(g_{l-1})) \\
&= \alpha_l(E_p[\log p_{g_{l-1}}] - E_{p_{g_{l-1}}}[\log p_{g_{l-1}}]) \\
&= \alpha_l(E_p[\log p_{g_{l-1}} - \log p] + E_p[\log p] \\
&- E_{p_{g_{l-1}}}[\log p_{g_{l-1}} - \log p] - E_{p_{g_{l-1}}}[\log p]) \\
&= \alpha_l(-D(p||p_{g_{l-1}}) - D(p_{g_{l-1}}||p) + E_p[\log p] - E_{p_{g_{l-1}}}[\log p]) \qquad (4.55)
\end{aligned}$$

Plugging back (4.55) into (4.54) yields:

$$\begin{aligned}
\min_{|\beta_l| \leq L, \theta_l} \alpha_l(E_{p_{g_{l-1}}}[\Delta_l] - E_p[\Delta_l]) &= \alpha_l(-D(p||p_{g_{l-1}}) - D(p_{g_{l-1}}||p) + E_p[\log p] - E_{p_{g_{l-1}}}[\log p]) \\
&+ \min_{|\beta_l| \leq L, \theta_l} \alpha_l(E_{p_{g_{l-1}}}[\beta_l \phi_{\theta_l}] - E_p[\beta_l \phi_{\theta_l}]) \qquad (4.56)
\end{aligned}$$

We use the ***mean value theorem*** *to bound* $\min_{|\beta_l| \leq L, \theta_l} Q(\theta_l, \beta_l)$ *where* $Q(\theta_l, \beta_l) = (E_{p_{g_{l-1}}}[\beta_l \phi_{\theta_l}] - E_p[\beta_l \phi_{\theta_l}])$. *Based on the mean value theorem*

$$\min_{|\beta_l| \leq L, \theta_l} Q(\theta_l, \beta_l) \leq \min_{|\beta_l| \leq L} E_\Pi(Q(\theta_l, \beta_l)), \qquad (4.57)$$

where

$$E_\Pi(Q(\theta_l, \beta_l)) = E_{p_{g_{l-1}}}[\int \beta_l \phi_{\theta_l} \Pi(\theta) d\theta] - E_p[\int \beta_l \phi_{\theta_l} \Pi(\theta) d\theta], \qquad (4.58)$$

and $\Pi = \tilde{\lambda}(\theta)\mu(d\theta)/\int \tilde{\lambda}(\theta)\mu(d\theta)$ *is a probability measure on* Θ*. Moreover*

$$\begin{aligned}\min_{|\beta_l|\leq L} E_\Pi(Q(\theta_l,\beta_l)) &\leq E_\Pi(Q(\theta_l,\beta')),\quad \beta' = \int \tilde{\lambda}(\theta)\mu(d\theta)\\ &\leq E_{p_{g_{l-1}}}[\int \phi_{\theta_l}\tilde{\lambda}(\theta)d\theta] - E_p[\int \phi_{\theta_l}\tilde{\lambda}(\theta)d\theta]\\ &\leq E_{p_{g_{l-1}}}[\log p] - E_p[\log p] \qquad (4.59)\end{aligned}$$

Thus,

$$\min_{|\beta_l|\leq L,\theta_l} \alpha_l(E_{p_{g_{l-1}}}[\beta_l\phi_{\theta_l}] - E_p[\beta_l\phi_{\theta_l}]) \leq \alpha_l(E_{p_{g_{l-1}}}[\log p] - E_p[\log p]). \qquad (4.60)$$

Plugging back (4.60) into (4.56) yields

$$\min_{|\beta_l|\leq L,\theta_l} \alpha_l(E_{p_{g_{l-1}}}[\Delta_l] - E_p[\Delta_l]) \leq \alpha_l(-D(p||p_{g_{l-1}}) - D(p_{g_{l-1}}||p)). \qquad (4.61)$$

If we substitute (4.61) into (4.53), then with probability at least $1-\delta$

$$G(\alpha_l) \leq \alpha_l K_1' + \alpha_l^2 K_2 + \alpha_l^3 K_3 - \alpha_l D(p_{g_{l-1}}||p). \qquad (4.62)$$

Since $\alpha_l D(p_{g_{l-1}}||p) \geq 0$*, then with probability at least* $1-\delta$

$$G(\alpha_l) \leq F(\alpha_l), \qquad (4.63)$$

where $F(\alpha_l) = \alpha_l K_1' + \alpha_l^2 K_2 + \alpha_l^3 K_3$. ■

Solve recursion error Given

$$E^{(l)} \leq \min_{0\leq\alpha_l\leq 1}(1-\alpha_l)E^{(l-1)} + \alpha_l K_1' + \alpha_l^2 K_2 + \alpha_l^3 K_3, \tag{4.64}$$

and $\alpha_l = \frac{3}{l+2}$, we have

$$E^{(l)} \quad \leq \quad K_1' + \frac{6K_2}{l+2} + \frac{27K_3}{(l+2)^2}. \tag{4.65}$$

Proof *See Appendix 4.5.3.* ■

Error bound in step m By setting $l = m$ and $m = \sqrt{n}$ in (4.65) for the error in step m we have

$$E^{(m)} \quad \leq \quad \frac{K_1\sqrt{d+2\log m}}{m} + \frac{6K_2}{m+2} + \frac{27K_3}{(m+2)^2}, \tag{4.66}$$

where $d = 2\log\frac{2}{\delta}$.

4.4.2.2 Left hand side inequality

We want to show that

$$E^{(m)} \geq -\frac{K_1\sqrt{d+2\log m}}{m}. \tag{4.67}$$

Proof *We start with the definition of* $E^{(l)}$

$$E^{(l)} = \min_{0\leq\alpha_l\leq 1,\beta_l,\theta_l} D(p||p_{g_l}) + (E_p[g_l] - E_{\hat{p}}[g_l]). \tag{4.68}$$

Since $\min_{0\leq\alpha_l\leq 1,|\beta_l|\leq L,\theta_l} D(p||p_{g_l}) \geq 0$, *thus*

$$E^{(l)} \geq \min_{0\leq\alpha_l\leq 1,|\beta_l|\leq L,\theta_l} (E_p[g_l] - E_{\hat{p}}[g_l]) \tag{4.69}$$

Using the Hoeffding inequality with probability at least $1-\delta$ *(see Appendix 4.5.2) as follows:*

$$-|E_p[g_l] - E_{\hat{p}}[g_l]| \geq -\frac{K_1\sqrt{d+2\log m}}{\sqrt{n}}. \tag{4.70}$$

By setting $m = \sqrt{n}$ *and plugging back (4.70) into (4.69), we have*

$$E^{(l)} \geq -\frac{K_1\sqrt{d+2\log m}}{m}. \tag{4.71}$$

Evaluating $E^{(l)}$ *at* $l = m$

$$E^{(m)} \geq -\frac{K_1\sqrt{d+2\log m}}{m}. \tag{4.72}$$

■

4.5 Appendix

4.5.1 proof of $\min_{\underline{\lambda},\underline{\theta}} D(p||p(x;\underline{\lambda}^*(\theta),\underline{\theta})) \leq 27(e^{2LM} - 1 - 2LM - 14/9L^2M^2)$

Let $g_0 = 0$. We construct g_l^* recursively as follows:

$$g_l^* = g_l(\alpha_l^*, \beta_l^*, \theta_l^*), \tag{4.73}$$

where

$$g_l = (1-\alpha_l)g_{l-1} + \alpha_l\beta_l\phi_{\theta_l}, \tag{4.74}$$

and α_l^*, β_l^*, and θ_l^* are chosen by

$$\alpha_l^*, \beta_l^*, \theta_l^* = \arg \min_{0\leq\alpha_l\leq 1, |\beta_l|\leq L, \theta_l} -E_p[\log p_{g_l}]. \tag{4.75}$$

Let $A_l = \min_{0\leq\alpha_l\leq 1,|\beta_l|\leq L,\theta_l} D(p||p_{g_l})$, where $p_{g_l} = e^{g_l - Z(g_l)}$, and $Z(g_l) = \log \int e^{g_l} dx$. We want to show that

$$A_l \leq \frac{6K_2}{l+2} + \frac{27K_3}{(l+2)^2}, \tag{4.76}$$

where $K_2 = 2L^2M^2$, and $K_3 = e^{2LM} - 1 - 2LM - 2L^2M^2$.

Proof *We start with A_l as follows:*

$$A_l = \min_{0\leq\alpha_l\leq 1,|\beta_l|\leq L,\theta_l} D(p||p_{g_l})$$

$$
\begin{aligned}
&= \min_{0\le\alpha_l\le 1,|\beta_l|\le L,\theta_l} E_p[\log p] + Z(g_l) - E_p[g_l] \\
&= \min_{0\le\alpha_l\le 1,|\beta_l|\le L,\theta_l} E_p[\log p] + Z(g_{l-1}+\alpha_l\Delta_l) - E_p[g_{l-1}+\alpha_l\Delta_l],\ \Delta_l = \beta_l\phi_{\theta_l} - g_{l-1} \\
&= \min_{0\le\alpha_l\le 1,|\beta_l|\le L,\theta_l} E_p[\log p] - E_p[g_{l-1}] + Z(g_{l-1}) + Z(g_{l-1}+\alpha_l\Delta_l) - Z(g_{l-1}) - \alpha_l E_p[\Delta_l] \\
&= \min_{0\le\alpha_l\le 1,|\beta_l|\le L,\theta_l} D(p||p_{g_{l-1}}) + \log E_{p_{g_{l-1}}}[e^{\alpha_l\Delta_l}] - \alpha_l E_p[\Delta_l] \\
&= A_{l-1} + \min_{0\le\alpha_l\le 1,|\beta_l|\le L,\theta_l} \log E_{p_{g_{l-1}}}[e^{\alpha_l\Delta_l}] - \alpha_l E_p[\Delta_l]. \qquad (4.77)
\end{aligned}
$$

We use the Taylor series $\log(1+\epsilon)\le\epsilon$ *to bound the log term. Thus*

$$
A_l \le A_{l-1} + \min_{0\le\alpha_l\le 1,|\beta_l|\le L,\theta_l} (E_{p_{g_{l-1}}}[e^{\alpha_l\Delta_l}] - 1) - \alpha_l E_p[\Delta_l]. \qquad (4.78)
$$

For $\Delta_l \le 2LM$*, we can bound* $e^{\alpha_l\Delta_l} \le 1 + \alpha_l\Delta_l + \alpha_l^2\Delta_l^2/2 + \alpha_l^3\frac{C_3|\Delta_l^3|}{6}$*, where* $C_3 = \frac{e^{2LM}-1-2LM-2L^2M^2}{4/3L^3M^3}$*. Thus,*

$$
\begin{aligned}
A_l &\le A_{l-1} + \min_{0\le\alpha_l\le 1,|\beta_l|\le L,\theta_l} E_{p_{g_{l-1}}}[\alpha_l\Delta_l + \alpha_l^2\Delta_l^2/2 + \alpha_l^3\frac{C_3|\Delta_l^3|}{6}] - \alpha_l E_p[\Delta_l] \\
&\le A_{l-1} + \min_{0\le\alpha_l\le 1}(\alpha_l^2 K_2 + \alpha_l^3 K_3 + \alpha_l \min_{|\beta_l|\le L,\theta_l} E_{p_{g_{l-1}}}[\Delta_l] - E_p[\Delta_l]) \qquad (4.79)
\end{aligned}
$$

where $K_2 = E_{p_{g_{l-1}}}[\Delta_l^2/2] \le 2L^2M^2$*, and* $K_3 = E_{p_{g_{l-1}}}[\frac{C_3|\Delta_l^3|}{6}] \le e^{2LM} - 1 - 2LM - 2L^2M^2$*. By expanding*

$$
E_{p_{g_{l-1}}}[\Delta_l] - E_p[\Delta_l] = E_{p_{g_{l-1}}}[g_{l-1}] - E_p[g_{l-1}] + E_p[\beta_l\phi_{\theta_l}] - E_{p_{g_{l-1}}}[\beta_l\phi_{\theta_l}], \qquad (4.80)
$$

we can reorganize $E_{p_{g_{l-1}}}[g_{l-1}] - E_p[g_{l-1}]$ *as:*

$$
\begin{aligned}
E_{p_{g_{l-1}}}[g_{l-1}] - E_p[g_{l-1}] &= (E_p[g_{l-1} - Z(g_{l-1})] + Z(g_{l-1}) - E_{p_{g_{l-1}}}[g_{l-1} - Z(g_{l-1})] - Z(g_{l-1})) \\
&= \alpha_l(E_p[\log p_{g_{l-1}}] - E_{p_{g_{l-1}}}[\log p_{g_{l-1}}])
\end{aligned}
$$

$$
\begin{aligned}
&= \alpha_l(E_p[\log p_{g_{l-1}} - \log p] + E_p[\log p] - E_{p_{g_{l-1}}}[\log p_{g_{l-1}} \\
&- \log p] - E_{p_{g_{l-1}}}[\log p]) \\
&= \alpha_l(-D(p||p_{g_{l-1}}) - D(p_{g_{l-1}}||p) + E_p[\log p] - E_{p_{g_{l-1}}}[\log p]) \qquad (4.81)
\end{aligned}
$$

Thus,

$$
\begin{aligned}
A_l &\leq A_{l-1} + \min_{0\leq\alpha_l\leq 1}(\alpha_l^2 K_2 + \alpha_l^3 K_3 + \alpha_l(-D(p||p_{g_{l-1}}) - D(p_{g_{l-1}}||p) + E_p[\log p] - E_{p_{g_{l-1}}}[\log p]) \\
&+ \alpha_l \min_{|\beta_l|\leq L,\theta_l} E_p[\beta_l\phi_{\theta_l}] - E_{p_{g_{l-1}}}[\beta_l\phi_{\theta_l}]) \qquad (4.82)
\end{aligned}
$$

We use the ***mean value theorem*** *to bound* $\min_{|\beta_l|\leq L,\theta_l} Q(\theta_l,\beta_l)$ *where* $Q(\theta_l,\beta_l) = (E_{p_{g_{l-1}}}[\beta_l\phi_{\theta_l}] - E_p[\beta_l\phi_{\theta_l}])$. *Based on the mean value theorem*

$$
\min_{|\beta_l|\leq L,\theta_l} Q(\theta_l,\beta_l) \leq \min_{|\beta_l|\leq L} E_\Pi(Q(\theta_l,\beta_l)), \qquad (4.83)
$$

where

$$
E_\Pi(Q(\theta_l,\beta_l)) = E_{p_{g_{l-1}}}[\int \beta_l\phi_{\theta_l}\Pi(\theta)d\theta] - E_p[\int \beta_l\phi_{\theta_l}\Pi(\theta)d\theta], \qquad (4.84)
$$

and $\Pi = \tilde{\lambda}(\theta)\mu(d\theta)/\int\tilde{\lambda}(\theta)\mu(d\theta)$ *is a probability measure on* Θ. *Moreover*

$$
\begin{aligned}
\min_{|\beta_l|\leq L} E_\Pi(Q(\theta_l,\beta_l)) &\leq E_\Pi(Q(\theta_l,\beta')), \quad \beta' = \int\tilde{\lambda}(\theta)\mu(d\theta) \\
&\leq E_{p_{g_{l-1}}}[\int\phi_{\theta_l}\tilde{\lambda}(\theta)d\theta] - E_p[\int\phi_{\theta_l}\tilde{\lambda}(\theta)d\theta] \\
&\leq E_{p_{g_{l-1}}}[\log p] - E_p[\log p] \qquad (4.85)
\end{aligned}
$$

Thus,

$$\min_{|\beta_l|\leq L,\theta_l} \alpha_l(E_{p_{g_{l-1}}}[\beta_l\phi_{\theta_l}] - E_p[\beta_l\phi_{\theta_l}]) \leq \alpha_l(E_{p_{g_{l-1}}}[\log p] - E_p[\log p]). \tag{4.86}$$

If we plug back (4.86) into (4.82) therefore,

$$A_l \leq \min_{0\leq\alpha_l\leq 1}(1-\alpha_l)A_{l-1} + \alpha_l^2 K_2 + \alpha_l^3 K_3 - \alpha D(p_{g_{l-1}}||p_{g_l}). \tag{4.87}$$

Since $-\alpha_l D(p_{g_{l-1}}||p_{g_l}) \leq 0$, *thus*

$$A_l \leq \min_{0\leq\alpha_l\leq 1}(1-\alpha_l)A_{l-1} + \alpha_l^2 K_2 + \alpha_l^3 K_3. \tag{4.88}$$

If we solve it for α_l *(see Section 4.5.1.1), then we have*

$$A_l \leq \frac{6K_2}{l+2} + \frac{27K_3}{(l+2)^2}. \tag{4.89}$$

■

4.5.1.1 Solve the recursion

Given

$$A_l \leq \min_{0\leq\alpha_l\leq 1}(1-\alpha_l)A_{l-1} + \alpha_l^2 K_2 + \alpha_l^3 K_3, \tag{4.90}$$

and $\alpha_l = \frac{3}{l}$, we have

$$A_l \leq \frac{6K_2}{l+2} + \frac{27K_3}{(l+2)^2}. \tag{4.91}$$

Proof *By induction, we show that if* $A_{l-1} \leq \frac{6K_2}{l+1} + \frac{27K_3}{(l+1)^2}$, *then* $A_l \leq \frac{6K_2}{l+2} + \frac{27K_3}{(l+2)^2}$. *We start with*

$$\begin{aligned} A_l &\leq (1 - \frac{3}{l+2})A_{l-1} + \frac{9K_2}{(l+2)^2} + \frac{27K_3}{(l+2)^3} \\ &\leq (1 - \frac{3}{l+2})(\frac{6K_2}{l+1} + \frac{27K_3}{(l+1)^2}) + \frac{9K_2}{(l+2)^2} + \frac{27K_3}{(l+2)^3}. \end{aligned} \tag{4.92}$$

We have to show that:

$$\begin{aligned} &(1 - \frac{3}{l+2})(\frac{6K_2}{l+1} + \frac{27K_3}{(l+1)^2}) + \frac{9K_2}{(l+2)^2} + \frac{27K_3}{(l+2)^3} \leq \frac{6K_4}{l+2} + \frac{27K_3}{(l+2)^2} \\ &\frac{l-1}{l+2}(\frac{6K_2}{l+1} + \frac{27K_3}{(l+1)^2}) + \frac{9K_2}{(l+2)^2} + \frac{27K_3}{(l+2)^3} \leq \frac{6K_2(l+2) + 27K_3}{(l+2)^2} \\ &\frac{l-1}{l+2}(\frac{6K_2(l+1) + 27K_3}{(l+1)^2}) + \frac{9K_2(l+2) + 27K_3}{(l+2)^3} \leq \frac{6K_2(l+2) + 27K_3}{(l+2)^2} \end{aligned} \tag{4.93}$$

The LHS can be simplified further as follows:

$$(6l^4 + 33l^3 + 54l^2 + 21l - 6)K_2 + (27l^3 + 108l^2 + 54l - 81)K_3.$$

The RHS also can be simplified as follows:

$$(6l^4 + 36l^3 + 78l^2 + 72l + 24)K_2 + (27l^3 + 108l^2 + 135l + 54)K_3.$$

Therefore, if we subtract LHS and RHS we get

$$-3K_2l^3 - 24K_2l^2 - (51K_2 + 91K_3)l - 165K_3, \tag{4.94}$$

which is always negative for $l \geq 1$. ■

4.5.2 Proof of $p(|\sum_{l=1}^{m} \lambda_l(\bar{\phi_{\theta_l}}(x) - E_p[\phi_{\theta_l}(x)])| \geq \frac{ML}{\sqrt{n}}\sqrt{2\log\frac{2m}{\delta}}) \leq \delta$

Define $X_l = \lambda_l(\bar{\phi_{\theta_l}}(x) - E_p[\phi_{\theta_l}(x)])$. We want to proof that

$$p(|\sum_{l=1}^{m} X_l| \geq \epsilon) \leq \delta, \tag{4.95}$$

where $\epsilon = \frac{ML}{\sqrt{n}}\sqrt{2\log\frac{2m}{\delta}}$.

Proof *We start with the LHS of (4.95) as follows:*

$$\begin{aligned}
&p(|\sum_{l=1}^{m} X_l| \geq \epsilon) \leq p(\sum_{l=1}^{m} |X_l| \geq \epsilon) \quad \textit{triangle inequality} \\
&p(\sum_{l=1}^{m} |X_l| \geq \epsilon) \leq p(\cup_{l=1}^{m} |X_l| \geq \epsilon) \\
&p(\cup_{l=1}^{m} |X_l| \geq \epsilon) \leq \sum_{l=1}^{m} p(|X_l| \geq \epsilon_l), (\forall \epsilon_l \geq 0, \sum_{l=1}^{m} \epsilon_l = \epsilon) \quad \textit{union bound}
\end{aligned} \tag{4.96}$$

Specifically we choose $\epsilon_l = \frac{||\lambda_l||_1 M\sqrt{2\log\frac{2}{\delta_l}}}{\sqrt{n}}$, *where* $||\phi_l||_\infty \leq M$. *Thus, if we show that*

$p(|X_l| \geq \frac{||\lambda_l||_1 M\sqrt{2\log\frac{2}{\delta_l}}}{\sqrt{n}}) \leq \delta_l$, *then based on (4.96)*

$$\begin{aligned} p(|\sum_{l=1}^{m} X_l| \geq \epsilon) &\leq \sum_{l=1}^{m} p(|X_l| \geq \epsilon_l) \\ &\leq \sum_{l=1}^{m} \delta_l = \delta. \end{aligned} \tag{4.97}$$

To prove $p(|X_l| \geq \frac{||\lambda_l||_1 M\sqrt{2\log\frac{2}{\delta_l}}}{\sqrt{n}}) \leq \delta_l$, *we proceed as follows:*

$$p(|X_l| \geq \epsilon_l) = p(|\lambda_l \sum_{i=1}^{n} (\phi_l(x_i) - E[\phi_l(x)])| \geq n\epsilon_l). \tag{4.98}$$

Because $p(-||\lambda_l||_1 M \leq |\lambda_l \phi_l(x_i)| \leq ||\lambda_l||_1 M) = 1$, *by applying Hoeffding inequality:*

$$p(|\lambda_l(\bar{\phi_l}(x_i) - E[\phi_l(x)]|) \leq 2e^{\frac{-2n^2\epsilon_l^2}{n(2M||\lambda_l||_1)^2}} = \delta_l \tag{4.99}$$

If we choose $\epsilon_l = \frac{||\lambda_l||_1}{||\lambda||_1}\epsilon$, *therefore*

$$\delta = \sum_{l=1}^{m} \delta_l = 2me^{\frac{-n\epsilon^2}{2(ML)^2}}. \tag{4.100}$$

Thus $\frac{\delta}{m} = 2e^{\frac{-n\epsilon^2}{2(ML)^2}} = \delta_l$, *and* $\epsilon = \frac{ML}{\sqrt{n}}\sqrt{2\log\frac{m}{2\delta}}$. ■

4.5.3 proof of $E^{(l)} \leq K_1' + \frac{6K_2}{l+2} + \frac{27K_3}{(l+2)^2}$

Consider the following recursion:

$$E^{(l)} \leq \min_{\alpha}(1-\alpha)E^{(l-1)} + \alpha K_1' + \alpha^2 K_2 + \alpha^3 K_3. \tag{4.101}$$

where $K_1' = K_1\frac{\sqrt{d+2\log m}}{m}$, $d = 2\log\frac{2}{\delta}$, $K_1 = LM$, $K_2 = 2L^2M^2$, $K_3 = e^{2LM} - 1 - 2LM - 2L^2M^2 > 0$. We want to show that

$$E^{(l)} \leq K_1' + \frac{6K_2}{l+2} + \frac{27K_3}{(l+2)^2}. \tag{4.102}$$

Proof *By induction, we show that if* $E^{(l-1)} \leq K_1' + \frac{6K_2}{l+1} + \frac{27K_3}{(l+1)^2}$, *then* $E^{(l)} \leq K_1' + \frac{6K_2}{l+2} + \frac{27K_3}{(l+2)^2}$. *We start with (4.101)*

$$\begin{aligned} E^{(l)} &\leq (1-\frac{3}{l+2})E^{(l-1)} + \frac{9K_2}{(l+2)^2} + \frac{27K_3}{(l+2)^3} \\ &\leq (1-\frac{3}{l+2})(K_1' + \frac{6K_2}{l+1} + \frac{27K_3}{(l+1)^2}) + \frac{3K_1'}{l+2} + \frac{9K_2}{(l+2)^2} + \frac{27K_3}{(l+2)^3}. \end{aligned} \tag{4.103}$$

We have to show that:

$$\begin{aligned} &(1-\frac{3}{l+2})(K_1' + \frac{6K_2}{l+1} + \frac{27K_3}{(l+1)^2}) + \frac{3K_1'}{l+2} + \frac{9K_2}{(l+2)^2} + \frac{27K_3}{(l+2)^3} \leq K_1' + \frac{6K_4}{l+2} + \frac{27K_3}{(l+2)^2} \\ &\frac{l-1}{l+2}(K_1' + \frac{6K_2}{l+1} + \frac{27K_3}{(l+1)^2}) + \frac{3K_1'}{l+2} + \frac{9K_2}{(l+2)^2} + \frac{27K_3}{(l+2)^3} \leq \frac{K_1'(l+2)^2 + 6K_2(l+2) + 27K_3}{(l+2)^2} \\ &\frac{l-1}{l+2}(\frac{K_1'(l+1)^2 + 6K_2(l+1) + 27K_3}{(l+1)^2}) + \frac{3K_1'(l+2)^2 + 9K_2(l+2) + 27K_3}{(l+2)^3} \leq \\ &\frac{K_1'(l+2)^2 + 6K_2(l+2) + 27K_3}{(l+2)^2} \end{aligned} \tag{4.104}$$

The LHS can be simplified further as follows:

$$(l^5+8l^4+25l^3+38l^2+28l+8)K_1'+(6l^4+33l^3+54l^2+21l-6)K_2$$
$$+(27l^3+108l^2+54l-81)K_3.$$

The RHS also can be simplified as follows:

$$(l^5+8l^4+25l^3+38l^2+28l+8)K_1'+(6l^4+36l^3+78l^2+72l+24)K_2$$
$$+(27l^3+108l^2+135l+54)K_3.$$

Therefore, if we subtract LHS and RHS we get

$$-3K_2l^3-24K_2l^2-(51K_2+91K_3)l-165K_3, \tag{4.105}$$

which is always negative for $l \geq 1$. ■

4.5.3.1 Bound for the error in step one

We want to show that

$$E^{(1)} \leq K_1' + 2K_2 + 3K_3. \tag{4.106}$$

Proof *Using (4.37) for error in step one we have:*

$$E^{(1)} = \min_{|\beta_1|\leq L,\theta_1} D(p||p_{g_1}) + (E_p[g_1] - E_{\hat{p}}[g_1]), \tag{4.107}$$

where $g_1 = \beta_1 \phi_{\theta_1}$. *Note that* $g_0 = 0$ *and* $\alpha_1 = 1$. *We can follow the same procedure as we did in Section 4.4.2.1 to bound* $E^{(1)}$. *Note that* $p_{g_{l-1}} = p_{g_0}$ *is the uniform distribution* p_u *which is defined as*

$$p_u = \begin{cases} \frac{1}{\int_{\mathcal{X}} dx} & x \in \mathcal{X} \\ 0 & o.w. \end{cases} \tag{4.108}$$

where $\mathcal{X}$ *is the support of* $p(x)$. *By applying Hoeffding inequality to the second term in the RHS of (4.107), we have*

$$E^{(1)} \leq K_1' + \min_{|\beta_1| \leq L, \theta_1} D(p||p_{g_1}) \tag{4.109}$$

where $K_1' = \frac{LM\sqrt{d+2\log m}}{\sqrt{n}}$, $d = 2\log\frac{2}{\delta}$. *Thus,*

$$\begin{aligned} \min_{|\beta_1| \leq L, \theta_1} D(p||p_{g_1}) &= \min_{|\beta_1| \leq L, \theta_1} E_p[\log p] + Z(g_1) - E_p[g_1] \\ &= \min_{|\beta_1| \leq L, \theta_1} E_p[\log p] + \log \frac{\int e_1^{\Delta} p_u}{p_u} - E_p[\Delta_1], \quad \Delta_1 = \beta_1 \phi_{\theta_1} \\ &= \min_{|\beta_1| \leq L, \theta_1} E_p[\log p] + \log E_{p_u}[e_1^{\Delta}] - E_{p_u}[\log p_u] - E_p[\Delta_1] \end{aligned} \tag{4.110}$$

We can bound $\log E_{p_u}[e_1^{\Delta}]$ *by applying the Taylor series to the log and exponent term (see (4.47), and (4.48)). Therefore,*

$$\min_{|\beta_1| \leq L, \theta_1} D(p||p_{g_1}) \leq K_2' + K_3' + E_p[\log p] - E_{p_u}[\log p_u] + \min_{|\beta_1| \leq L, \theta_1} E_{p_u}[\Delta_1] - E_p[\Delta_1], \tag{4.111}$$

where $K_2' = \frac{K_2}{4}$, *and* $K_3' = e^{LM} - 1 - LM - L^2M^2/2$. *If we use the same idea of the*

mean value theorem (see (4.57)) thus,

$$\min_{|\beta_1|\leq L,\theta_1} E_{p_u}[\Delta_1] - E_p[\Delta_1] \leq E_{p_u}[\log p] - E_p[\log p] \tag{4.112}$$

Plugging back (4.112) into (4.111) yields

$$\begin{aligned} \min_{|\beta_1|\leq L,\theta_1} D(p||p_{g_1}) &\leq K_2' + K_3' + E_p[\log p] - E_{p_u}[\log p_u] + E_{p_u}[\log p] - E_p[\log p] \\ &\leq K_2' + K_3' - D(p_u||p). \end{aligned} \tag{4.113}$$

Knowning that $-D(p_u||p) \leq 0$, if we plug back (4.113) into (4.109) yields

$$E^{(1)} \leq K_1' + K_2' + K_3'. \tag{4.114}$$

Since $E^{(1)} \leq K_1' + K_2' + K_3'$ and since $K_i \geq 0$, thus

$$E^{(1)} \leq K_1' + 2K_2 + 3K_3. \tag{4.115}$$

■

Chapter 5: Confidence-Constrained Maximum Entropy Framework for Learning Multi-instance Data

5.1 Introduction

Multi-instance learning (MIL) refers to a class of learning problem where each object represented as a bag of instances. For example, a document (bag) comprises of words (instance), an image (bag) consists of local region patches (instance), and a webpage (bag) is a list of links (instance). MIL has been applied to many areas in machine learning and signal processing e.g., drug activity detection [42], text classification [3, 124], object detection in image [109], and content-based image categorization [36, 123]. Machine learning algorithms are described as either *supervised* or *unsupervised*. Multi-instance learning (MIL) refers to the prediction problem or supervised learning [3, 36, 42, 112] in which the main goal is to predict the label of an unseen bag, given the label information of the training bags. On the other hand, learning from multi-instance data in an unsupervised manner is called grouped data modeling [22, 23, 107] in which the main goal is to uncover the underlying (hidden) structure of the data in the input. In the supervised MIL, each bag is associated with a class of label and the goal is to predict an unseen bag given all instances inside the bag. Due to the ambiguity of the label information related to instances and weak association between instance-level information and bag-level information, supervised MIL is a challenging task. Since the introduction of the MIL approach in machine learning and signal processing, numerous

algorithms have been proposed either by upgrading traditional algorithms to MIL e.g., citation kNN [112], MI-SVM and mi-SVM [3], neural network MIL [96], or devising a new algorithm specifically for MIL e.g., axis-parallel rectangles (APR) [42], diverse density (DD) [80], EM-DD [122], and MIBoosting [115]. MIL has been studied in an unsupervised setting in [60, 119, 121].

In all of the above mentioned algorithms, the instance-level similarity metric has been used such as Hausdorff or Mahalabonis distance [65, 111, 112, 116]. The instance-level metrics are computationally expensive and increases quadratically in the number of instances in each bag [116]. Moreover, instance level metrics cannot reflect the structure similarity defined in the bag level and it is difficult to identify the characteristic of each bag using instance-level similarity [52]. For example, images with similar objects in some regions and many other incompatible objects in other regions could not be identified in the same class using the instance-level metric [111]. Some kernel approaches have been proposed which consider the statistical properties of the instances to measure the similarity in the bag-level [52]. Each bag is mapped to a single point, then a kernel is used to classify at the bag-level. The problem of computational complexity associated with instance-level metrics has been solved by trying to represent each bag with few samples in a very high dimension e.g., single-blob-with-neighbors (SBN) representation for each image [81]. Moreover, this abstract representation can avoid significant effects of noise in each bag. That is because in labeling each bag, it is positive if and only if one instance inside the bag is positive. Thus, having a rich representation of each bag may introduce some noise in each bag which can not be captured by instance-level similarity. A statistical representation of each bag can address this problem.

In this work, we consider the problem of associating each bag with a probability distribution obtained by the principle of maximum entropy. Assuming that each instance

in a bag is generated *i.i.d.* from an unknown density function, we fit to each bag a density function exploiting the statistical property of instances which can capture the structure of the data at the bag level. This approach has several advantages over existing approaches. First, the problem can be solved in a convex framework. Second, it maps each bag of instances into a point in the probability distribution space which captures the structure of the data. In this framework each bag is parameterized by a vector which carries all the information about the instances inside each bag. This approach brings the problem of multi-instance learning from instance-level into bag-level where it is convenient to learn a meaningful metric and computationally is less complex. Third, a meaningful metric can be defined over the space of distributions to measure the similarity among bags. Moreover, the computational complexity significantly drops from quadratically in the number of instances to quadratically in the number of features.

Our contributions in this work are: 1) we introduce a new framework for MIL using the principle of maximum entropy approach, 2) a metric defined over the space of the distributions is introduce to measure the similarities among bags in MIL, 3) we proposed confidence-constrained maximum entropy to learn the space of distributions jointly, 4) an accelerated proximal gradient approach is proposed to solve the convex optimization problem, 5) the performance of the proposed approach is evaluated in terms of rank recovery in the space of distributions and compared with regularized MaxEnt, and 6) we examined the classification accuracy of CCMaxEnt on four real world dataset and compared the results with the state-of-the-art algorithms in MIL.

5.2 Problem statement

Suppose we are given a set of N bags $\{X_1, X_2, \ldots, X_N\}$, where $X_i \in \mathcal{X}$. The instances in the bag are denoted as $\{x_{i1}, x_{i2}, \ldots, x_{in_i}\}$, where n_i is the total number of instance in bag i. We consider the problem of unsupervised learning of distribution for each bag using the maximum entropy framework. The goal is to 1) provide a latent representation for each bag X_i using a generative model p_{λ_i} obtained by maximum entropy, 2) provide a joint probability framework with some regularization which takes into account the model complexity and insufficient number of samples, 3) provide a framework to examine the accuracy of the estimation obtained by the proposed framework. The representation as a distribution (instead of a set of points) allows bags to be represented in the same space, i.e., $p_{\lambda_i} \in \mathcal{P}$. This approach provides a framework where structure can be introduced at the probability distribution level rather than at the instance level. For example, dimension reduction can be performed in the distribution space $\mathcal{P}$ rather than in the instance space $\mathcal{X}$.

Mapping each bag to a distribution using the maximum entropy approach provides an abstraction in data representation. In this representation, each density p_{λ_i} correspond to one bag X_i. Thus, the similarity measurement between two bags X_i and X_j is equivalent to measuring the distance between the corresponding densities p_{λ_i} and p_{λ_j} in the space of the probability. We use KL-divergence between two densities p_{λ_i} and p_{λ_j}. The complexity for computing bag-level similarity measures after summarizing each bag by a statistic is superior to that of instance-level similarity based methods such as Hausdorff distance.

5.3 Maximum entropy framework for multi-instance data

We consider the maximum entropy framework for modeling multiple-instance datasets by treating multi-instance examples as probability distributions. We are interested in the development of a framework that will allow convenient incorporation of structure (e.g., geometric, low-dimension) in the distribution space. The problem of density estimation can be define as follows. Given an *i.i.d.* set of samples $X = \{x_1, x_2, \ldots, x_n\}$ from an unknown density function p, find an estimator for p. We use the framework of maximum entropy to estimate p [2]. In the maximum entropy framework one is interested in identifying a unique distribution given a set of constraints on generalized moments of the distributions: $E_p[\phi_j] = \alpha_j$ where $\phi \in \mathbb{R}^m$ is feature function defined over the space of the samples. The basis functions ϕ summarize the statistical property of samples by mapping each sample into a single point. It is assumed that the unknown distribution p can be parametrized by a set of coefficients $\lambda \in \mathbb{R}^m$, and that an estimate of these parameters can be obtained by solving a convex optimization problem. This framework has the advantage of not restricting the class of the distribution to a specific density and considers a wide range of density functions in the class of exponential family. In fact, it is shown that with a rich set of basis function ϕ, the approximation error decreases in order of $\mathcal{O}(1/m)$ where m is the number of basis [15]. We explain the maximum entropy approach below.

5.3.1 Single density estimation (SDE)

The maximum entropy (MaxEnt) framework for density estimation was first proposed by Janes [64] and has been used in many areas of computer science and signal process-

ing including natural language processing [18, 40], species distribution modeling [47, 92], text classification [87, 125], and image processing [102] . The maximum entropy framework [37] finds a unique probability density function (p.d.f) over $\mathcal{X}$ that satisfies the constraints $E_p[\phi(x)] = \alpha$, where $\phi(x) \in \mathbb{R}^m$ is feature transformation defined over $\mathcal{X}$. In principle, many p.d.f.'s can satisfy the constraints. The maximum entropy approach selects a unique distribution among them. The problem of single density estimation in the maximum entropy framework can be formulated as:

$$\begin{aligned} &\text{maximize} \quad H(p) \\ &\text{subject to} \quad E_p[\phi_j] = E_{\hat{p}}[\phi_j] \\ &\qquad\qquad \int p(x)dx = 1, \end{aligned} \tag{5.1}$$

where $H(p) = -\int p(x) \log p(x) dx$ is the entorpy of $p(x)$, $E_p[\phi_j] = \int p(x)\phi_j dx$ and $E_{\hat{p}}[\phi_j] = \frac{1}{n}\sum_{l=1}^{n} \phi_j(x_l)$ is the empirical mean of $\phi(x)$. It can be shown that a solution to (5.1) can be represented as follows:

$$p_\lambda(x) = \exp\big(\lambda^T \phi(x) - Z(\lambda)\big), \tag{5.2}$$

where $Z(\lambda) = \log \int \exp \lambda^T \phi(x)$. We will now derive the maximum-likelihood (ML) estimator for the parameter λ in p_λ given n *i.i.d.* observations $x_1, \ldots, x_n$. First, note that assuming the form of p.d.f. in (5.2), the log likelihood can be written as

$$\mathcal{L} = \log p(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} (\lambda^T \phi(x_i) - Z(\lambda)) = n(\lambda^T E_{\hat{p}}[\phi(x)] - Z(\lambda)). \tag{5.3}$$

Note that $\sum_{i=1}^{n} \phi(x_i) = nE_{\hat{p}}[\phi(x)]$. Thus, we can write the negative log-likelihood function as follows:

$$\begin{aligned} -\mathcal{L} &= -nE_{\hat{p}}[\lambda^T \phi(x) - Z(\lambda)] \\ &= nE_{\hat{p}}[\log \hat{p}] - nE_{\hat{p}}[\lambda^T \phi(x) - Z(\lambda)] \\ &= nD(\hat{p} \| p_\lambda) + \Upsilon, \end{aligned} \tag{5.4}$$

where $\Upsilon = nE_{\hat{p}}[\log \hat{p}]$ is a constant and $D(p\|q) = E_p[\log p - \log q]$. Therefore, maximizing the log-likelihood in (5.3) w.r.t. λ is equivalent to minimizing the KL-divergence in (5.4) w.r.t. λ. Thus, $\hat{\lambda}$ can be obtained as a result of the following optimization problem:

$$\begin{aligned} \hat{\lambda} &= \arg\min_{\lambda} nD(\hat{p} \| p_\lambda) \\ &= \arg\min_{\lambda} n(Z(\lambda) - \lambda^T E_{\hat{p}}[\phi(x)]), \end{aligned} \tag{5.5}$$

where $\bar{\phi} = \sum_{l=1}^{n} \phi(x_l)/n \in \mathbb{R}^m$. There are several algorithms for solving MaxEnt e.g., iterative scaling [40] and its variants [47,92], gradient descent, Newton, and quasi-Newton approach [78, 99]. The ML optimization problem is convex in terms of λ and can be solved efficiently using Newton's method. Newton's method requires the first and second derivative of the objective function w.r.t. λ. These derivatives are given below.

$$\begin{aligned} \nabla_\lambda &= n(E_{p(\lambda)}[\phi] - \bar{\phi}) \\ \nabla^2_\lambda &= n(E_{p(\lambda)}[\phi] E_{p(\lambda)}[\phi]^T - E_{p(\lambda)}[\phi\phi^T]) \end{aligned} \tag{5.6}$$

Algirthm 5 lists Newton's method. MaxEnt can overfit data due to low number of samples or large number of feature function ϕ [46, 47]. Regularized MaxEnt is proposed

Algorithm 5 Single density estimation algorithm

Input: $X = \{x_1, x_2, \ldots, x_n\}$ sample from bag X, K, $\phi \in \mathbb{R}^m$, $\lambda^0 \in \mathbb{R}^m$.
Output: $\lambda \in \mathbb{R}^m$ and Z
for $k = 1$ to K **do**
  $\Delta\lambda^k = -{\nabla^2_{\lambda^k}}^{-1}\nabla^k_\lambda$
  Find t^k using backtracking
  $\lambda^{k+1} = \lambda^k + t^k\Delta\lambda^k$
end for

to overcome the issue of overfitting in MaxEnt [34, 46, 56, 67, 70]. Regularized MaxEnt can be either formulated as relaxing the equality in (5.1) [34, 56] or putting a prior on the p.d.f. in (5.2) [34, 67] (Laplace prior yields l_1 regularization and Gaussian prior yields l_2 regularization). Algorithms for solving regularized MaxEnt are proposed in [34,46,47,56,67]. Convergence analysis for regularized MaxEnt is provided in [15,46,47]. The problem of single density estimation is presented to introduce the maximum entropy framework for density estimation. In the next section, we show how to use the maximum entropy framework for multiple density estimation in MIL.

5.3.2 Multiple density estimation (MDE)

Multiple density estimation (MDE) for MIL can be done following the same principle as explained for single density estimation in the previous section. In MDE each bag is represented by one distribution and the cost function for MDE, due to bag independence, is the sum of the cost functions for all bags. MDE can be solved using the following minimization:

$$\hat{\underline{\lambda}} \quad = \quad \arg\min_{\underline{\lambda}} \sum_{i=1}^{N} n_i D(\hat{p}_i \| p_{\lambda_i})$$

$$= \quad \arg\min_{\underline{\lambda}} \sum_{i=1}^{N} n_i (Z(\lambda_i) - \lambda_i^T \bar{\phi}_i), \tag{5.7}$$

where $\hat{\underline{\lambda}} = [\hat{\lambda}_1, \ldots, \hat{\lambda}_N]$ and $\underline{\lambda} = [\lambda_1, \ldots, \lambda_N]$. MDE formulation proposed in (5.7) considers the density estimation for each bag individually. This individual estimate addresses the nature of each dataset separately and ignores the fact that the underlying structure of the data can be shared among all datasets. This might cause a poor generalization performance due to the low number of samples for some datasets [45]. One the other hand, we can pool data and considers all the data comes from one density which in fact ignores the important differences among the datasets. A middle ground approach is to use regularization to force the joint density estimation while keep the origin of each data uninfluenced. Hierarchical density estimation [46] formulates the problem of MDE using l_1 regularization. The regularization defined on each data separately and on the group of the data defined in the hierarchy. Note that the hierarchal structure of the data is a prior information. However, in most cases in the real world applications the relations among the datasets are unknown beforehand, e.g., in text or image datasets. In the following, we proposed a framework for learning jointly in the space of distributions using the principle of maximum entropy.

5.3.3 Rank recovery in the space of distributions

In this section, we introduce the concept of rank recovery in the space of distributions. Later, we show how rank recovery can help in jointly learning the space of distributions. The dimension of the space of distributions is controlled by the size of the basis $\phi = [\phi_1, \phi_2, \ldots, \phi_m]^T$. Often the size of ϕ is large to allow accurate approximation of the

distribution space. Hence, we are interested in finding a smaller basis that provides a fairly accurate replacement to the original basis ϕ. We consider the problem of finding a new basis in the span of ϕ. Suppose a smaller basis ψ can be obtained by $\psi = A^T\phi$, where $\psi = [\psi_1, \psi_2, \ldots, \psi_k]^T$ and A is a $m \times k$ matrix, where $k < m$. Instead of using $\lambda_i^T\Phi$ involving m terms, one can use $\beta_i^T\psi$ involving only k terms. In this case, $\phi^T\Lambda = \psi^T\beta = \phi^T A\beta$, where $\Lambda = [\lambda_1, \lambda_2, \ldots, \lambda_N]$ and $\lambda_i \in \mathbb{R}^m$, which results in $\Lambda = A\beta$ such that $A \in \mathbb{R}^{m\times k}$ and $\beta \in \mathbb{R}^{k\times N}$. Hence $\Lambda = A\beta$ is a low-rank matrix.

5.3.4 Regularized MDE (RegMDE) using MaxEnt

To obtain a low-rank solution for Λ, we can solve a regularized nuclear norm MDE. The nuclear norm of a matrix $\|X\|_*$ is defined as the sum of the singular values of matrix X. The nuclear norm is a special class of Schatten norm which is defined as $\|X\|_p = (\sum_i \sigma_i^p)^{\frac{1}{p}}$. When $p = 1, \|X\|_p$ is equal to the nuclear norm. Nuclear norm enforces sparsity on the singular value of matrix X which results in low-rank structure. The heuristic replacement of rank with nuclear norm has been proposed for various application such as matrix completion [29, 97], collaborative filtering [103], and multi-task learning [93].

In RegMDE, a regularized nuclear norm is added to the objective function in (5.7) yielding:

$$\text{minimize} \qquad \sum_{i=1}^{N} n_i(Z(\lambda_i) - \lambda_i^T\bar{\phi}_i) + \eta\|\Lambda\|_*, \tag{5.8}$$

where η is the regularization parameter. RegMDE can be viewed as maximum a posteriori (MAP) criterion using a prior distribution over matrix Λ of the form $Ce^{-\eta\|\Lambda\|_*}$.

This is similar to the interpretation of l_1-regularization for sparse recovery as MAP with a Laplacian prior. A quasi-Newton approach has been proposed to solve RegMDE [9]. RegMDE can also be formulated as a constrained MDE as follows:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{N} n_i(Z(\lambda_i) - \lambda_i^T \bar{\phi}_i), \\ \text{subject to} \quad & \|\Lambda\|_* \leq \nu, \end{aligned} \tag{5.9}$$

where $\nu \geq 0$ is a tuning parameter. For each value of η in (5.8) there is a value of ν in (5.9) which produces the same solution [54]. One of the main challenges in regularized and constrained ML is the choice of regularization parameters η and ν, respectively. Often, a criterion for selection a value for regularization parameter that guarantees exact rank recovery is unavailable. There is an extensive discussion in [14] for exact rank recovery in regularized and constrained MDE. We propose the concept of confidence-constrained rank minimization for jointly learning the space of distributions which overcome the issues of parameter tuning with regularized and constrained MDE.

5.3.5 Confidence-constrained MaxEnt (CCMaxEnt)

We propose confidence-constrained MaxEnt for learning the space of distributions jointly. Using the properties of the maximum entropy framework, an in-probability bound on the objective function in (5.7) can be obtained. The probability bound on the log-likelihood function allows us to define a confidence set. A confidence set is a high-dimensional generalization of the confidence interval which we use to restrict the search space of the problem. Search inside the confidence set guarantees a low-rank solution. Hence, in this approach the roles of ML objective and rank constraint are reversed. We consider rank

minimization subject to ML objective constraint. The CCMaxEnt is given by:

$$\begin{aligned} &\text{minimize} && \text{Rank}(\Lambda) \\ &\text{subject to} && \sum_{i=1}^{N} n_i D(p_{\hat{\lambda}_i} \| p_{\lambda_i}) \leq \epsilon(\delta), \end{aligned} \tag{5.10}$$

where $\epsilon(\delta)$ is an in-probability bound for the estimation error. Note in this formulation the tuning parameter $\epsilon(\delta)$ can be obtained by bounding $\sum_{i=1}^{N} n_i D(p_{\hat{\lambda}_i} \| p_{\lambda_i})$ (see Appendix 5.7.1). Since (5.10) involves rank minimization which is non-convex, we provide an alternative convex relaxation to (5.10) in the following.

5.3.6 CCMaxEnt nuclear norm minimization

In general, rank minimization problems are NP hard [82]. Various algorithms have been proposed to solve the general rank minimization problem locally (e.g., see [58, 83]). A heuristic replacement of the rank minimization with a nuclear norm minimization is commonly proposed [50, 97]. To solve the rank minimization problem proposed in (5.10), we propose the widely used approach of replacing the rank minimization with the tractable convex optimization problem of nuclear norm minimization. In the following, CCMaxEnt nuclear norm minimization is proposed as a convex alternative to (5.10):

$$\begin{aligned} &\text{minimize} && \|\Lambda\|_* \\ &\text{subject to} && \sum_{i=1}^{N} n_i D(p_{\hat{\lambda}_i} \| p_{\lambda_i}) \leq \epsilon. \end{aligned} \tag{5.11}$$

We denote the solution to (5.11) by $\hat{\Lambda}_*$. Since the nuclear norm is a convex function, and the set of the inequality and equality constraints construct a convex set, (5.11)

is a convex optimization problem. This nuclear norm regularization encourages a low-rank representation to feature space, i.e., all features can be represented as a linear combination of a few alternative features. Assume $\Lambda = USV^T = \sum_j u_j s_j v_j^T$ is the singular value decomposition of Λ, then

$$\begin{aligned} \lambda_i^T \phi(x) &= \sum_{j=1}^{k} s_j (e_i^T v_j)(u_j^T \phi(x)) \\ &= \sum_{j=1}^{k} s_j (e_i^T v_j)\psi_j(x) = \beta_i^T \psi(x) \end{aligned} \tag{5.12}$$

where k is the rank of matrix Λ. Similar to principle component analysis, where each data point can be approximated as a linear combination of a few principle components, each bag can be represented as a distribution using a linear combination of only a few basis functions $\psi_1(x), \ldots, \psi_k(x)$. This method facilitates a dimension reduction in the space of distributions.

5.4 Proximal gradient approach to solve CCMaxEnt nuclear norm minimization

The optimization problem in (5.11) can be written as follows:

$$\begin{aligned} &\text{minimize} \quad f(\Lambda) \\ &\text{subject to} \quad g(\hat{\Lambda}, \Lambda) \leq \epsilon, \end{aligned} \tag{5.13}$$

where $f(\Lambda) = \|\Lambda\|_*$ and $g(\hat{\Lambda}, \Lambda) = \sum_{i=1}^{N} D(p_{\hat{\lambda}_i} \| p_{\lambda_i}) n_i$. The Lagrangian dual of (5.13) is

$$L(\Lambda, z) = f(\Lambda) + z(g(\hat{\Lambda}, \Lambda) - \epsilon), \tag{5.14}$$

where $z \geq 0$ is the dual variable. Given $\nabla g(\hat{\Lambda}, \Lambda)$ is Lipschitz continuous with parameter $\tau_g = Nm^2$ (see Appendix), where N is total number of bags and m is total number of feature functions, a quadratic upper bound for (5.14) can be written as:

$$\begin{aligned} f(\Lambda) + z(g(\hat{\Lambda}, \Lambda) - \epsilon) &\leq f(\Lambda) + z(g(\hat{\Lambda}, \Lambda_0) + (\Lambda - \Lambda_0)^T \nabla g(\hat{\Lambda}, \Lambda_0) + \frac{\tau_g}{2} \|\Lambda - \Lambda_0\|_F^2 - \epsilon) \\ &= \|\Lambda\|_* + z(\frac{\tau_g}{2} \|\Lambda - \Lambda'\|_F^2 - \frac{1}{2\tau_g} \nabla g(\hat{\Lambda}, \Lambda_0)^2 - \epsilon) \\ &= Q(\Lambda, \Lambda_0), \end{aligned} \tag{5.15}$$

where $\Lambda' = \Lambda_0 - \frac{1}{\tau_g} \nabla g(\hat{\Lambda}, \Lambda_0)$. $Q(\Lambda, \Lambda_0)$ is a quadratic bound on the Lagrangian $\mathcal{L}$. We consider minimizing $Q(\Lambda, \Lambda_0)$ w.r.t. Λ due to its closed form solution. The solution to the minimization of $Q(\Lambda, \Lambda_0)$ w.r.t. Λ is

$$\hat{\Lambda}_*(z) = \mathcal{D}_{\frac{1}{\tau_g z}}(\Lambda') \tag{5.16}$$

where $\mathcal{D}_\alpha(X)$ is the soft-thresholding operator on the singular values of matrix X (for proof see [27]) defined by $D_\alpha(X) = U(S - \alpha I)_+ V^T$, where $X = USV^T$ is the SVD of X. To find z^* we have to maximize $Q(\hat{\Lambda}_*(z), \Lambda_0)$ w.r.t. z. Since parameter z is a scalar, we propose a greedy search approach to find the optimum z.

5.4.1 step size

In the proximal gradient approach, Λ will be updated in each iteration based on $1/\tau_g$. In fact, $1/\tau_g$ plays the role of step size. However, in practice it is usually very conservative to set a constant step size τ_g [108]. As long as the inequality $L(\Lambda, z) \leq Q(\Lambda, \Lambda_0)$ is hold, the step size can be increased. Therefore, a linesearch-like algorithm is proposed to find a smaller value for τ_g which satisfies the inequality.

5.4.2 Acceleration

The convergence rate for the proximal gradient approach is $\mathcal{O}(1/k)$ where k is the number of iteration [74, 108]. The convergence rate of the gradient approach can be speed up to $\mathcal{O}(1/k^2)$ using the extrapolation technique proposed in [86] given the fact that the gradient is Lipschitz continuous. In our problem, the gradient of $g(\hat{\Lambda}, \Lambda)$ is Lipschitz continuous with $\tau_g = Nm^2$ where N is total number of bags and m is total number of features.

The only costly part of the proximal algorithm is the evaluation of the singular values in each iteration. Note that in each iteration of soft-thresholding operator we need to know the number of singular values greater than a threshold. As in [14, 27, 73, 108], we use the PROPACK package to compute a partial SVD. To accelerate the proximal gradient approach for CCMaxEnt, we use the acceleration technique proposed in [86]. The pseduo code for CCMaxEnt nuclear norm minimization is proposed in Algorithm 6.

Algorithm 6 CCMaxEnt nuclear norm minimization

Input: $X_i = \{x_{i1}, x_{i2}, \ldots, x_{in_i}\}$ sample from bag X_i, $i = 1, \ldots, N$, K, $\phi \in \mathbb{R}^m$, $\Lambda^1, \Lambda^0 \in \mathbb{R}^{m \times N}$,$a_1 = a_0 = 1$, z_-^1, z_+^1, and $\alpha \in (0, 1)$.
Output: $\lambda_i^* \in \mathbb{R}^m$ and $Z(\lambda_i^*)$
for $j = 1$ to $\ldots$ **do**
  $z^k = \frac{z_-^k + z_+^k}{2}$ {Dual variable update}
  for $k = 1$ to K **do**
    $\bar{\Lambda} = \Lambda^k + \frac{a^{k-1}-1}{a^k}(\Lambda^k - \Lambda^{k-1})$
    while $L(\bar{\Lambda}^k, z^k) \leq Q(\bar{\Lambda}^k, \bar{\Lambda}^{k-1})$ **do**
      $\tau_g^k = \alpha \tau^{k-1}$
    end while
    $G^k = \Lambda^k - \frac{1}{\tau_g^k} \nabla g(\hat{\Lambda}, \bar{\Lambda}^k)$ {Proximal step}
    Compute $\Lambda^{k+1} = D_{\frac{z^k}{\tau}}(G^k)$ {proximal update}
    $a^{k+1} = \frac{1+\sqrt{1+4a^k}}{2}$
  end for
  {Line search for dual variable z}
  if $g(\Lambda) - \epsilon \geq 0$ **then**
    $z_-^{k+1} = z_k$
  else
    $z_+^{k+1} = z_k$
  end if
  if $\sum_i D(p_{\hat{\lambda}_i} \| p_{\lambda_i}) n_i - \epsilon < consTol$ **then**
    break
  end if
end for

5.5 Experiments

In this section, we evaluate both theoretical and computational aspect of CCMaxEnt compare to RegMDE for rank recovery in the space of distributions. For the theoretical part we provide a phase diagram analysis to evaluate the performance of both CCMaxEnt and RegMDE in rank recovery. We then provide an illustration of distribution space dimension reduction using CCMaxEnt and RegMDE. Moreover, we show that CCMaxEnt introduces a metric which can be used in object similarity recognition in image

processing.

5.5.1 Phase diagram analysis

We use the notion of phase diagram [44] to evaluate probability of rank recovery using CCMaxEnt and RegMDE for a wide range of matrices Λ of different dimensions (i.e., features size $\times$ number of bags) and different number of topics (rank of matrix Λ). We construct distributions using low-rank matrix Λ and draw $i.i.d$ samples using rejection sampling (data are generated in $2D$ space). Figure 5.1 shows the contour plot of the first 4 distributions used in our experiment. For the random samples drawn from the constructed distributions, we obtain $\hat{\Lambda}$ by maximum likelihood estimation (e.g., see (5.7)). Note that $\hat{\Lambda}$ is a noisy version of matrix Λ and is full rank. We consider two

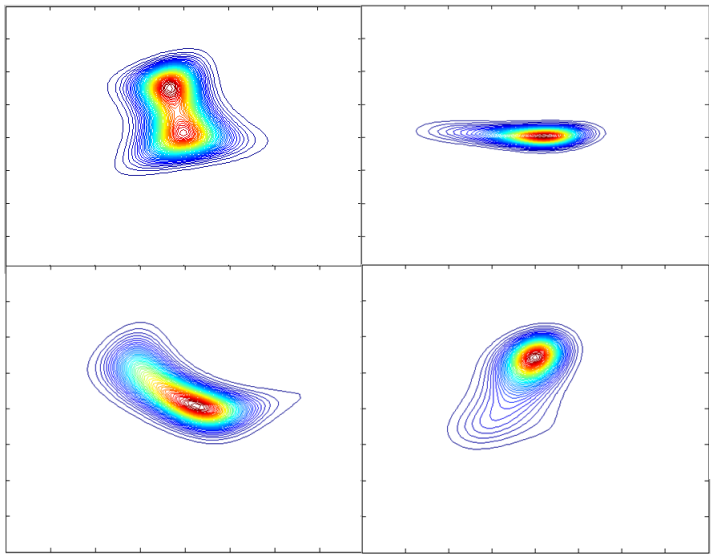


Figure 5.1: Contour plot of the first 4 distributions used in our experiment

different setups for number of bags: $N = 50$ and $N = 500$. We would like to illustrate the performance of CCMaxEnt and RegMDE in small ($N = 50$) and large ($N = 500$)

scale problems in terms of rank recovery. For $N = 50$ bags, we vary the number of features and number of topics (rank of matrix Λ) over a grid of (m, T) with m (number of features) ranging through 7 equispaced points in the interval $[20, 50]$ and T (rank of matrix Λ) ranging through 10 equispaced points in the interval $[2, 20]$ (see Fig. 5.2). Each pixel intensity in phase diagram corresponds to the empirical evaluation of the probability of exact rank recovery. For each pixel in the phase diagram we produce 10 realization of $\hat{\Lambda}$ (each $\hat{\Lambda}$ is obtained using rejection sampling and then maximum likelihood estimation). We run CCMaxEnt and RegMDE for each of 10 realization of $\hat{\Lambda}$ and compare the rank of the obtained matrix Λ^* with the rank of the true Λ. The rank evaluation is done by counting the number of singular values of matrix Λ^* exceeding a threshold. The threshold is defined based on the empirical distribution of the smallest nonzero singular values of the true matrix Λ (i.e., mean minus three times the standard deviation). To find the regularization parameter η in RegMDE (5.8), we use a cross validation approach and continuation technique [77, 108]. The continuation technique in nuclear norm minimization is similar to the path following algorithm in solving l_1 regularized regression (LASSO) proposed in [49]. Convergence analysis of continuation technique is shown in [59]. For cross validation, we consider a range of regularization parameter $\eta = \{10^{-4}, 10^{-3}, \ldots, 10^3, 10^4\}$. For each value of η, we separate data into train and test sets (70% train and 30% test), and evaluate the test error using the objective function in (5.8), then select η^* as the value corresponding to the lowest test error. For continuation technique, we set η to a large value ($\eta^0 = \|\hat{\Lambda}\|_F^2$) and repeatedly solve the optimization problem (5.8) with a decreasing sequence of η^k until we reach the target value $\bar{\eta}$ ($\eta^k = \max(1e^{-1}\eta^{k-1}, \bar{\eta})$) where $\bar{\eta} = 1e^{-3}\eta^0$. Due to large value of η in the beginning of the algorithm, matrix Λ^* is low-rank and in each iteration we increase the rank of Λ^*. Note that the value of constant $1e^{-3}$ in $\bar{\eta} = 1e^{-3}\eta^0$ and $1e^{-1}$ in

$\eta^k = \max(1e^{-1}\eta^{k-1}, \bar{\eta})$ is set manually based on preliminary experiments. The stopping criteria for CCMaxEnt is the combination of $MaxIter \leq 100$, $objTol < 1e^{-2}$, and $consTol < 1e^{-1}$ where $MaxIter$ is the maximum number of iteration of main algorithm, $objTol$ is the tolerance of objective function $\|\mathrm{f}_{\min}^{k-1} - \mathrm{f}_{\min}^{k}\|_1$, and $consTol$ is the tolerance for violating the confidence constraint $\|\sum_i D(p_{\hat{\lambda}_i}\|p_{\lambda_i})n_i - \epsilon\|$. The stopping criteria for RegMDE is the same as for CCMaxEnt except that $objTol$ is not used. Figure 5.2(a), 5.2(b), and 5.2(c) show the phase diagram results for exact rank recovery with CCMaxEnt, RegMDE (cross validation), and RegMDE (continuation technique) for $N = 50$. The white

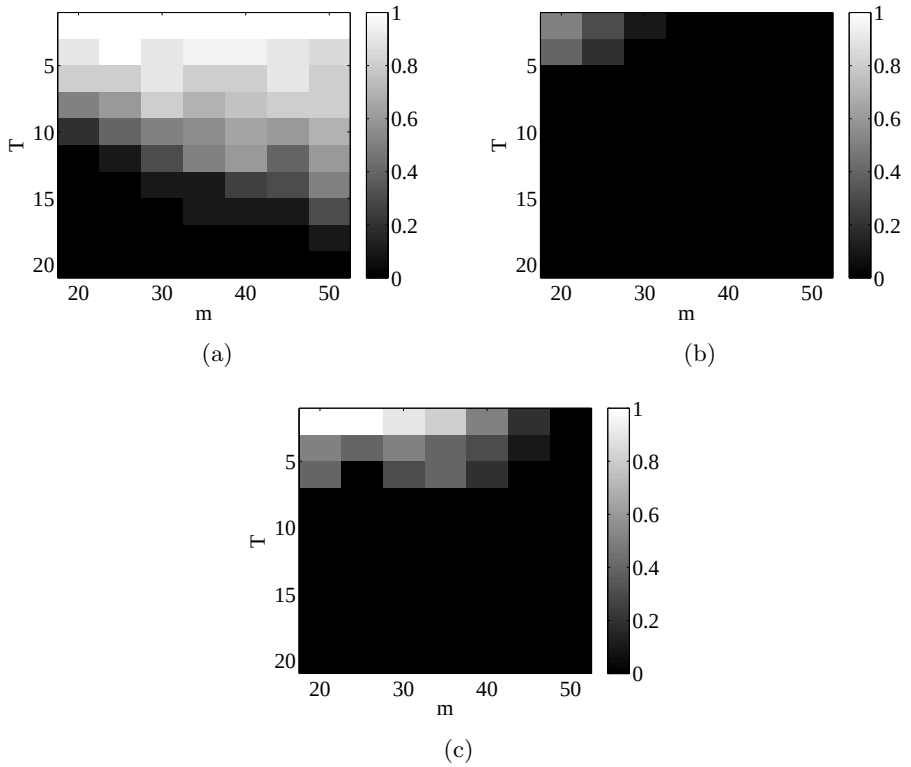


Figure 5.2: Comparison of probability of exact rank recovery obtained by (a) CCMaxEnt, (b) RegMDE with cross validation and (c) RegMDE with continuation technique for $N = 50$.

region in Fig. 5.2(a) and Fig. 5.2(b) correspond to the probability of exact rank recovery

obtained by CCMaxEnt and RegMDE, respectively. The white area in Fig. 5.2(a) is wider than the white areas in Fig. 5.2(b) and Fig. 5.2(c). This is because the proximal gradient method makes more progress per iteration than the L-BFGS algorithm, but both run for the same number of iterations. The class of L-BFGS algorithms is usually slow for non-smooth problems [117]. Moreover, in the proximal gradient approach used in CCMaxEnt, we use a quadratic bound on the main objective function which results in a closed-form expression for the proximal operator. Based on Eckart-Young [104] a low-rank matrix has the lower error in terms of quadratic cost function. Another observation is that the white area in RegMDE with continuation technique is slightly wider than RegMDE with cross validation technique. This could be due to the fact that in the continuation technique we start we a very low-rank matrix Λ and increase the rank gradually until we reach a targeted value, whereas in the cross validation technique we keep the regularization parameter constant throughout the optimization.

For $N = 500$, we scan the number of features and number of topics (rank of matrix Λ) over a grid of (m, T) with m ranging through 19 equispaced points in the interval $[100, 1000]$ and T ranging through 20 equispaced points in the interval $[5, 100]$. Due to the high computational complexity of scanning through different values of η in RegMDE with cross validation, and better result in terms of rank recovery in RegMDE with continuation technique on small scale data ($N = 50$), we compare rank recovery between CCMaxEnt and RegMDE with continuation technique in this case. Figure 5.3(a) and 5.3(b) show the phase diagram results for exact rank recovery with CCMaxEnt, RegMDE (cross validation), and RegMDE (continuation technique) for $N = 500$. We observe that the white area in CCMaxEnt approach is wider than the white areas in RegMDE approach.

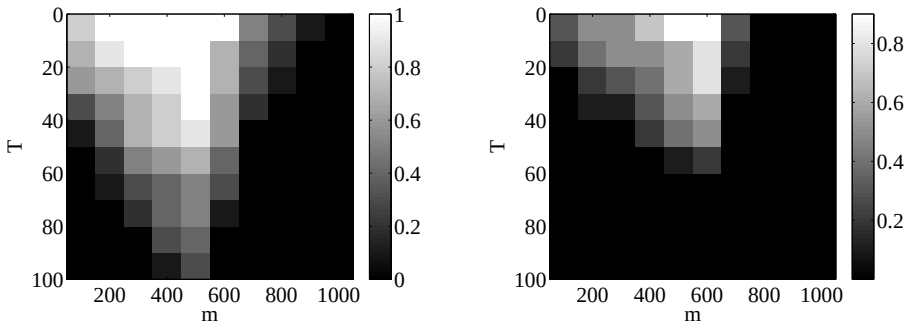


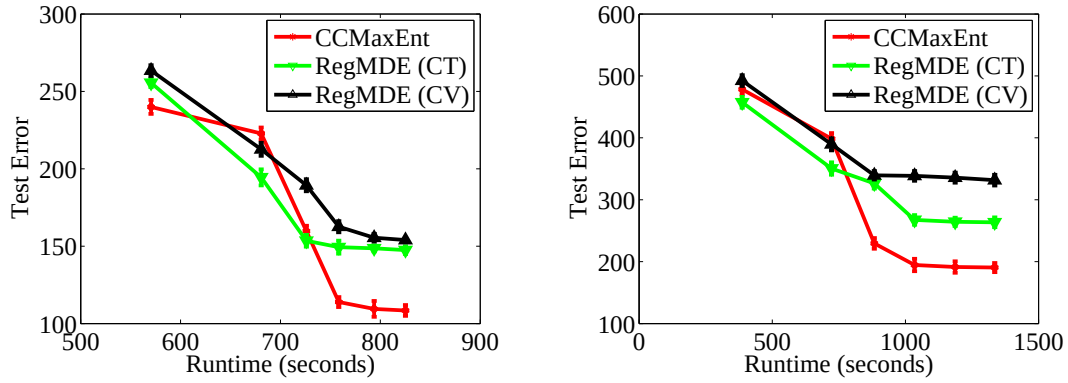
Figure 5.3: Comparison of probability of exact rank recovery obtained by (a) CCMaxEnt and (b) RegMDE with continuation technique for $N = 500$.

5.5.2 Parameter estimation error

We compare the test error vs. runtime for both CCMaxEnt and RegMDE on a synthetic dataset. We construct a low-rank matrix Λ and generate *i.i.d.* samples from the low-rank distribution and estimate matrix $\hat{\Lambda}$ using maximum likelihood estimation. Then we obtain matrix Λ^* using CCMaxEnt and RegMDE. We consider $N = 50$, $m = 100$, $T = 5$, and $T = 20$. We randomly choose 70% of the data as a training set and test on the rest of the data over 10 different realizations. The test error is evaluated as $\sum_i Z(\lambda_i) - \lambda_i^T \bar{\phi}$, where i indexes all bags in the test set. Figure 5.4 shows the results of test error vs. runtime [1]. Figure 5.4(a) shows the result for $T = 5$. Since initially finding the true model with correct rank in CCMaxEnt is computationally expensive (due to dual variable update), we observe that RegMDE is performing better than CCMaxEnt in the beginning. However, we see that overall the test error in CCMaxEnt decreases faster than RegMDE. In Fig. 5.4(b), the result is shown for $T = 20$. We see that by increasing the complexity of the model, it takes longer for CCMaxEnt and RegMDE to

[1] We run all algorithms on a standard desktop computer with 2.5 GHz CPU (dual core) and 4 GB of memory implemented in MATLAB.

find the correct model.



(a) (b)

Figure 5.4: Comparison of test error vs. runtime for $N = 50$, $m = 100$, and (a) $T = 5$, (b) $T = 20$.

5.5.3 Dimension reduction

The purpose of this section is to illustrate how dimension reduction can be achieved using the ψ obtained by CCMaxEnt. Since all the datasets are high dimensional, we use PCA as a preprocessing step. Figure 5.5 depicts the whole process of implementing our approach for one image in the Corel1000 dataset [48]. We use the block representation of the image followed by PCA to reduce the dimension. The image is represented as a bag of instances where each instance corresponds to a small rectangular patch of pixels. The feature vector describing each patch is the raw pixel intensities (RGB) with PCA applied to reduce the dimension. We perform the CCMaxEnt approach to learn a p.d.f. over the block representation of the image. After performing the nuclear norm minimization in (5.11) on the Corel1000 dataset, we select one image as an example. Then, we choose the first few bases of matrix ψ obtained by (5.12) to represent the image as a linear

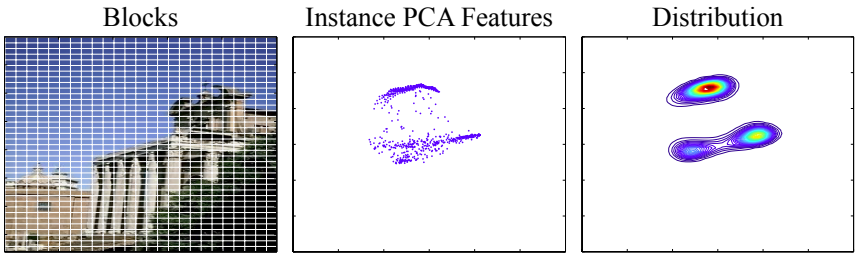


Figure 5.5: The whole MaxEnt process from bag representation to fitting a distribution. The figures from left to right shows the following: (1) how an images is represented as a bag of instances (blocks), (2) The $2D$ PCA features of each instance (3) the density fitted to the data using the maximum entropy principle.

combination of these basis functions. Figure 5.6 shows that the contour plots of these basis functions. To provide intuitive understanding, we name each basis ψ following the content of the image corresponding to instances near the peaks of ψ. The first column of Fig. 5.6 is an image and its corresponding estimated density. The other columns show each ψ_i and the part of the image that corresponds to that ψ_i.

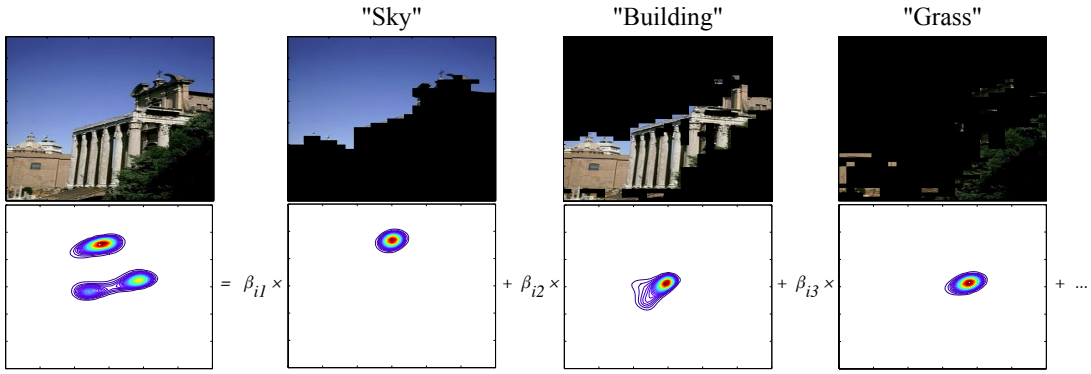


Figure 5.6: Dimension reduction in the space of the distribution obtained by the bases ψ. The first column shows the image and corresponding density estimation. The other columns show each ψ and part of the image that corresponds to that ψ.

5.5.4 KL-divergence similarity

For classification and retrieval, it is useful to have a similarity measure between bags. The Kullback-Leibler (KL) divergence between two estimated distributions provides such a similarity measures. The KL divergence between two distributions obtained by the maximum entropy approach has a closed form:

$$D(p_{\lambda_i} \| p_{\lambda_j}) = (\lambda_i - \lambda_j)^T E_{p_{\lambda_i}}[\Phi] - (Z_i - Z_j). \tag{5.17}$$

We symmetrize the divergence by adding $D(p_{\lambda_i} \| p_{\lambda_j}) + D(p_{\lambda_j} \| p_{\lambda_i})$.

$$D(p_{\lambda_i} \| p_{\lambda_j}) + D(p_{\lambda_j} \| p_{\lambda_i}) = (\lambda_i - \lambda_j)^T (E_{p_{\lambda_i}}[\Phi] - E_{p_{\lambda_i}}[\Phi]). \tag{5.18}$$

Figure 5.7 shows a set of images and their nearest images identified by KL-divergence similarity. We clearly observe that by using the KL-divergence similarity, the nearest neighbors are relevant to the main images which validates the efficacy of the proposed similarity measure.

Figure 5.7: Top: Query image. Bottom: Nearest-neighbor based on KL-divergence.

5.5.5 Application

We also evaluate the classification accuracy of the proposed KL-divergence based similarity measure when used in distance-based multi-instance algorithms such as Citation-kNN [112] and bag-level kernel SVM [52]. We compare KL-divergence to bag-level distance measures that rely on pairwise instance-level comparisons, namely average Hausdorff distance [112] and the RBF set kernel [52], both in terms of accuracy and runtime. The comparison is conducted over four datasets, i.e., the Corel1000 image dataset [48] Musk1, Musk2 [42], and Flowcytometry [31]. The Corel1000 [48] image dataset consists of 10 different classes each containing 100 images. We use 50 randomly subsampled images from 4 classes: *'buildings', 'buses', 'flowers', and 'elephants'*. We represent each image (bag) as a collection of instances, each of which corresponds to a 10×10 pixel block, and is described by a feature vector of all pixel intensities in 3 color channels (RGB). The Musk1 dataset [42] describes a set of 92 molecules of which 47 are judged by human expert to be musks and the remaining 45 molecules are judged to be non-musk. The Musk2 dataset [42] is a set of 102 molecules of which 39 are judged by human experts to be musks and the remaining 63 molecules are judged to be non-musks. Each instance corresponds to a possible configuration of a molecule. The Flowcytometry dataset consists of $5d$ vector reading of multiple blood cell samples for each one of 43 patients. For each patient, we have two similar cell characterstics with respect to the antigens surface which are called 1) chronic lymphocytic leukemia (CLL) or 2) mantle cell lymphoma (MCL). Each patients are considered as a bag and the blood samples are instances in the bag. Table 5.1 summarizes the properties of each dataset.

Table 5.1: Datasets

Dataset	bags	no. of class	Ave. inst/bag	dim
Corel1000 (4class)	200	4	950	300
Musk1	92	2	4.5	166
Musk2	102	2	64.7	166
Flowcytometry	43	2	5664	5

5.5.6 Experimental setup

We use classification accuracy as an evaluation metric. In all experiments, we use the preprocessed datasets obtained by PCA. We perform 10-fold cross validation over all datasets. As baselines, we implement a modified version of Citation-kNN [112] replacing the Hausdorff distance with KL-divergence, and a bag-level SVM with the kernel for two bags X and X' defined as $K(X, X') = e^{-\gamma D_{KL}(X,X')}$, $K(X, X') = e^{-\gamma D_{Haus}(X,X')}$, and the RBF set kernel used by [52]. Below we state the ranges of all tunning parameters for these algorithms used in our experiments. We compared CCMaxEnt with RegMDE with cross validation and RegMDE with continuation technique. We use a grid of $\{10^{-4}, 10^{-3}, \ldots, 10^{3}, 10^{4}\}$ for the regularization parameter η. All of the datasets use features with dimension reduced by PCA. We use a grid of $\{2, 3, 4, 5, 6, 7\}$ for the feature dimension after applying PCA. The Citation-kNN algorithm has two parameters- the number of nearest neighbors k, and the number of "citers" k'. We use a grid of $\{1, 5, 10, 15, 20\}$ for k and $\{5, 10, 15, 20, 25\}$ for k'. The SVM has two parameters- the bandwith of RBF kernel γ, and the penalty factor C. We use a grid of $\{2^{-9}, 2^{-8}, \ldots, 2^{0}\}$ for γ, and a grid of $\{2^{0}, 2^{1}, \ldots, 2^{9}\}$ for C. For the basis functions used in constructing the maximum entropy distribution space, we propose $\phi_{2k} = \cos(g_k^T x)$ and $\phi_{2k-1} = \sin(g_k^T x)$, where $g_k \sim \mathcal{N}(0, I)$ i.i.d for $k = 1, 2, \ldots, m/2$. In [94], a similar transformation is used to approximate Gaussian kernels.

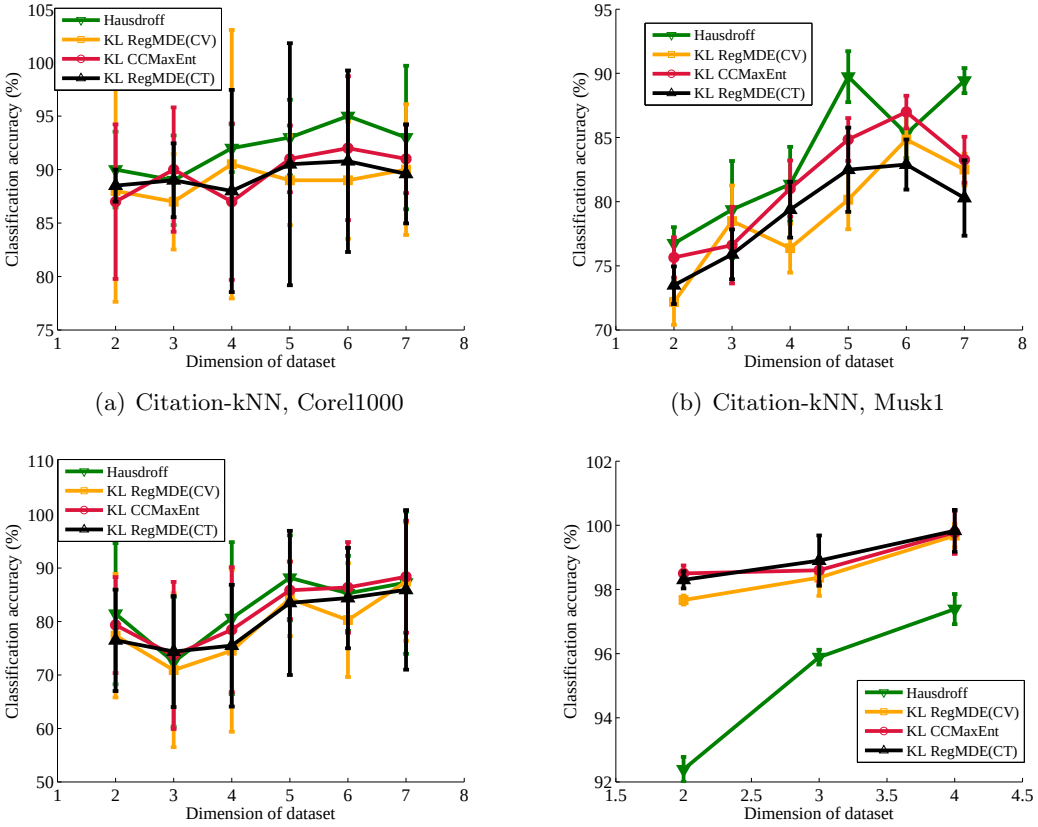
5.5.7 Classification accuracy experiments

The results of classification accuracy for citation-kNN for four datasets are shown in Fig. 5.8. We compared the classification accuracy with Citation-kNN using KL-divergence and Hausdorff distance. The KL divergence is computed from 3 different distribution estimates: 1) RegMDE(CV): regularized MDE with cross validation, 2) RegMDE(CT): regularized MDE with continuation, and 3) CCMaxEnt: confidence-constrained maximum entropy. We observe that CCMaxEnt performs slightly better than RegMDE in musk and image datasets where in Flowcytometry dataset RegMDE(CT) is performing better. However, the difference is not very significant. Overall, Hausdorff distance has better classification accuracy than the other approaches which can be due to measuring distance in the instance level. Figure 5.9 shows the results for bag-level SVM with the RBF set kernel, the average Hausdorff distance kernel, and the KL divergence kernel obtained by RegMDE and CCMaxEnt. In general, KL divergence is performing better than Hausdorff distance. Accuracy results are very close to all methods using KL divergence.

5.5.8 Runtime

To compare the computational complexity of our algorithm with standard MIL algorithms, we run Citation-kNN and MI-SVM using the MIL toolkit[2] on the Corel1000 image dataset for different numbers of instances in each bag. To evaluate how the runtime of each algorithm depends on the number of instances in the dataset, we randomly sample varying number of instance from each bag. In Fig. 5.10, the x-axis shows the

[2]http://www.cs.cmu.edu/ juny/MILL/



(a) Citation-kNN, Corel1000

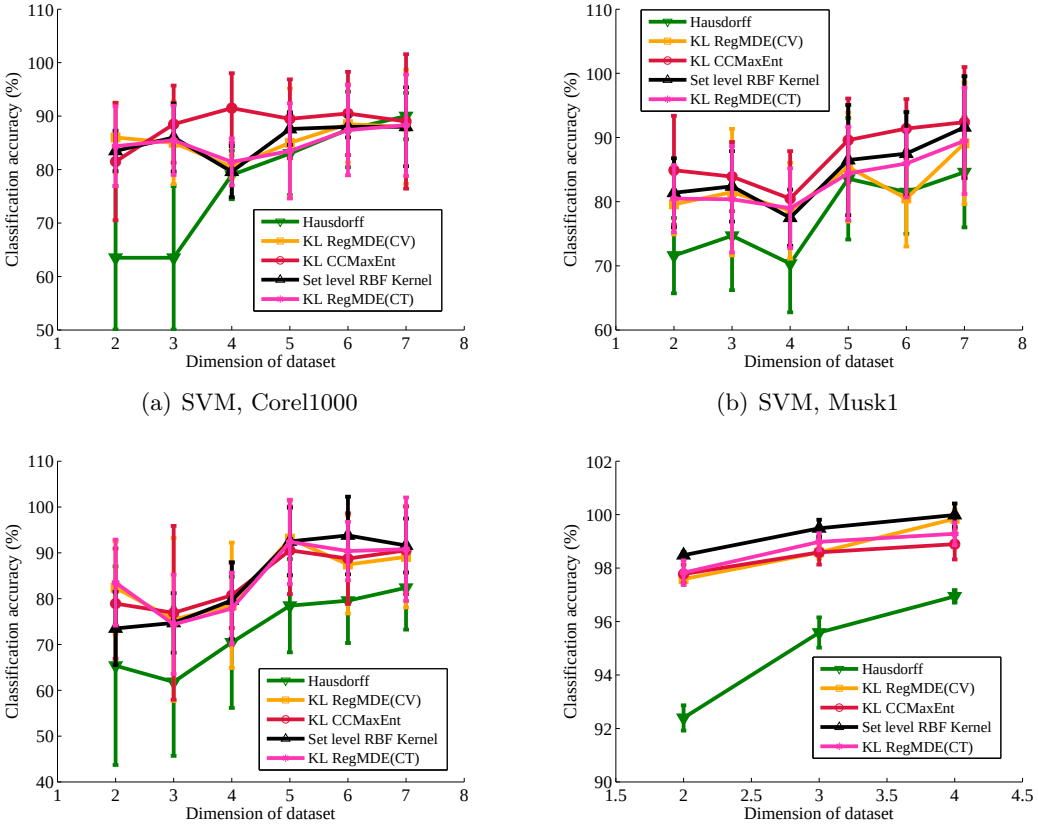
(b) Citation-kNN, Musk1

(c) Citation-kNN, Musk2

(d) Citation-kNN, Flowcytometry

Figure 5.8: Classification accuracy results for (a) Corel1000, (b) Musk1 (c) Musk2 and (d) Flowcytometry.

number of samples in each bag and the y-axis shows the elapsed CPU time in seconds. We compare the time complexity of standard MIL algorithm with RegMDE (CV), RegMDE (CT), and CCMaxEnt. The runtime of Citation-kNN and SVM is significantly longer than RegMDE and CCMaxEnt by several orders of magnitude. Hence our proposed approach achieves superior runtime and similar accuracy to two standard MIL algorithms. The computational complexity of RegMDE and CCMaxEnt during training is $\mathcal{O}(Nndm)$, where n is average number of instance per bag, d is the dimension of



(a) SVM, Corel1000

(b) SVM, Musk1

(c) SVM, Musk2

(d) SVM, Flowcytometry

Figure 5.9: Classification accuracy results for (a) Corel1000, (b) Musk2 (c) Musk1 and (d) Flowcytometry. Set level RBF kernel accuracy is provided for reference.

instances, and m is the number of basis functions ϕ and during test is $\mathcal{O}(Nm)$. The computational complexity of Hausdorff distance during test is $\mathcal{O}(n^2N)$. The Haussdorff distance based approach requires no training.

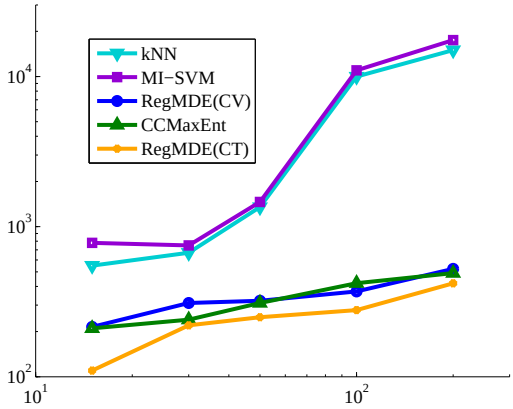


Figure 5.10: Time comparison among Citation-kNN, MI-SVM, RegMED, and CCMax-Ent.

5.5.9 Discussion

RegMDE and CCMaxEnt approach for MIL is significantly faster than other algorithms when there are a large number of instances in each bag. RegMDE and CCMaxEnt achieve this speedup by summarizing the instances in each bag, thereby avoiding instance-level processing in later steps. Moreover, using regularization helps in utilizing the information of other similar bags when constructing a density estimate.

5.6 Conclusion

In this work, we propose a confidence-constrained maximum entropy approach for multi-instance learning problem. The proposed approach used the idea of representing each bag in the space of distribution. This approach summarizes the high volume data in multi-instance learning utilizing the statistical properties of each bag. We proposed the framework of maximum entropy to fit density for each bag. Moreover, we introduce regularization for learning the space of distribution which is conveniently handled in this

framework.

5.7 Appendix

5.7.1 Proof of probability bound for $\sum_{i=1}^{N} D(p_{\hat{\lambda_i}} \| p_{\lambda_i}) n_i$

To find the probability bound for the random quantity $\sum_{i=1}^{N} D(p_{\hat{\lambda_i}} \| p_{\lambda_i}) n_i$, we use Markov's inequality. Markov's inequality for random variable X and a positive scalar a is defined as follows:

$$p(|X| \geq a) \leq \frac{E(X)}{a}. \tag{5.19}$$

In fact Markov's inequality relates the probability of random variable X to its expectation. Since $p(\sum_{i=1}^{N} D(p_{\hat{\lambda}_i} \| p_{\lambda_i}) n_i \geq 0) = 1$, we propose the following bound for the random quantity $\sum_{i=1}^{N} D(p_{\hat{\lambda}_i} \| p_{\lambda_i}) n_i$

$$p(\sum_{i=1}^{N} D(p_{\hat{\lambda}_i} \| p_{\lambda_i}) n_i \geq \epsilon(\delta_k)) \leq \frac{1}{k}, \tag{5.20}$$

where $\epsilon(\delta_k) = \frac{kNm}{2}$, N is the number of datasets, and m is the number of feature functions. To do so, we have to obtain the $E\left[\sum_{i=1}^{N} D(p_{\hat{\lambda}_i} \| p_{\lambda_i}) n_i\right]$. We first consider the quantity $D(p_{\hat{\lambda}_i} \| p_{\lambda_i})$ and expand it as follows:

$$D(p_{\hat{\lambda}_i} \| p_{\lambda_i}) \quad = \quad (\hat{\lambda}_i - \lambda_i)^T \bar{\phi}_i(x) - (Z(\hat{\lambda}_i) - Z(\lambda_i)) \tag{5.21}$$

Using the Taylor series expansion around λ_i in (5.21), an upper bound can be obtained

as follows:

$$D(p_{\hat{\lambda}_i}\|p_{\lambda_i}) \leq (\hat{\lambda}_i-\lambda_i)^T\bar{\phi}_i(x)-((\hat{\lambda}_i-\lambda_i)^T\dot{Z}(\lambda_i)+\frac{1}{2}(\hat{\lambda}_i-\lambda_i)^T\ddot{Z}(\lambda_i)(\hat{\lambda}_i-\lambda_i)).$$

Since $\bar{\phi}_i(x)=\dot{Z}(\hat{\lambda}_i)$, thus

$$\begin{aligned} D(p_{\hat{\lambda}_i}\|p_{\lambda_i}) &\leq (\hat{\lambda}_i-\lambda_i)^T(\dot{Z}(\hat{\lambda}_i)-\dot{Z}(\lambda_i))-\frac{1}{2}(\hat{\lambda}_i-\lambda_i)^T\ddot{Z}(\lambda_i)(\hat{\lambda}_i-\lambda_i) \\ &\leq (\hat{\lambda}_i-\lambda_i)^T\ddot{Z}(\lambda_i)(\hat{\lambda}_i-\lambda_i)-\frac{1}{2}(\hat{\lambda}_i-\lambda_i)^T\ddot{Z}(\lambda_i)(\hat{\lambda}_i-\lambda_i) \\ &\leq \frac{1}{2}(\hat{\lambda}_i-\lambda_i)^T\ddot{Z}(\lambda_i)(\hat{\lambda}-\lambda_i). \end{aligned} \tag{5.22}$$

Note that in the firs line of (5.22) we use the first order Taylor series expansion of $(\dot{Z}(\hat{\lambda}_i)-\dot{Z}(\lambda_i))$. We take the expectation of $D(p_{\hat{\lambda}_i}\|p_{\lambda_i})$ as follows:

$$\begin{aligned} E\Big[D(p_{\hat{\lambda}_i}\|p_{\lambda_i})\Big] &\leq \frac{1}{2}E\Big[(\hat{\lambda}_i-\lambda_i)^T\ddot{Z}(\lambda_i)(\hat{\lambda}-\lambda_i)\Big] \\ &= \frac{1}{2}E\Big[\mathrm{tr}\left(\ddot{Z}(\lambda_i)(\hat{\lambda}-\lambda_i)(\hat{\lambda}_i-\lambda_i)^T\right)\Big] \\ &= \frac{1}{2}\mathrm{tr}\left(\ddot{Z}(\lambda_i)E\Big[(\hat{\lambda}-\lambda_i)(\hat{\lambda}_i-\lambda_i)^T\Big)]\right) \\ &= \frac{1}{2}\mathrm{tr}\left(\ddot{Z}(\lambda_i)\mathrm{Cov}(\lambda_i)\right) \\ &= \frac{1}{2}\mathrm{tr}\left(\ddot{Z}(\lambda_i)\frac{\ddot{Z}(\lambda_i)^{-1}}{n_i}\right) \\ &= \frac{m}{2n_i}. \end{aligned} \tag{5.23}$$

Note that we used the fact that $\mathrm{Cov}(\lambda_i)=\frac{\ddot{Z}(\lambda_i)^{-1}}{n_i}$. Since $D(p_{\hat{\lambda}_i}\|p_{\lambda_i}), i=1,\ldots,m$ are

independent random variables, to obtain $E\left[\sum_{i=1}^{N} D(p_{\hat{\lambda}_i} \| p_{\lambda_i}) n_i\right]$ we can write

$$\begin{aligned} E\left[\sum_{i=1}^{N} D(p_{\hat{\lambda}_i} \| p_{\lambda_i}) n_i\right] &= \sum_{i=1}^{N} n_i E\left[D(p_{\hat{\lambda}_i} \| p_{\lambda_i})\right] \\ &\leq \sum_{i=1}^{N} \frac{m n_i}{2 n_i} \\ &= \frac{Nm}{2}. \end{aligned} \tag{5.24}$$

Therefore, using the Markov's inequality with probability δ_k where $\delta_k = \frac{1}{k}$ we have

$$\sum_{i=1} N D(p_{\hat{\lambda}_i} \| p_{\lambda_i}) n_i \geq \epsilon(\delta_k), \tag{5.25}$$

where $\epsilon(\delta_k) = \frac{kNm}{2}$.

5.7.2 Proof of Lipschitz continuity for $\nabla g(\hat{\Lambda}, \Lambda)$

In this section, we want to show that $\nabla g(\hat{\Lambda}, \Lambda)$ is Lipschitz continuous with constant $\tau_g = Nm^2$ where N is total number of bags and m is total number of feature functions. We prove that the Hessian matrix $\nabla^2 g(\hat{\Lambda}, \Lambda)$ is bounded which is stronger than Lipschitz continuity of the gradient $\nabla g(\hat{\Lambda}, \Lambda)$. The Hessian of $g(\hat{\Lambda}, \Lambda)$ is equivalent to the covariance of the feature functions ϕ. Thus,

$$\begin{aligned} \nabla^2 g(\hat{\Lambda}, \Lambda) &= E_{p_\lambda}(\phi\phi^T) - \left(E_{p\lambda}(\phi) E_{p\lambda}(\phi)^T\right) \\ &\leq E_{p_\lambda}(\phi\phi^T) \\ V^T E_{p_\lambda}(\phi\phi^T) V &\leq \int (V^T\phi)^2 p_\lambda dx \end{aligned}$$

$$\begin{aligned} &\leq \|V\|_2 \sum_i \phi_i^2 \\ &\leq Nm^2 \end{aligned} \tag{5.26}$$

Chapter 6: Conclusion

6.1 Contributions

In the following, we list a brief summary of our contributions in this dissertation. We categorize the contributions based on learning multi-instance data in the discrete and continuous domain. Specifically, in the discrete domain

1. We proposed sufficient conditions for exact rank recovery in topic models as a rank minimization problem and provided a new framework for parameter free confidence-constrained convex optimization as an alternative to rank minimization problem, which can overcome the issues of Bayesian inferences such as *i*) computational complexity associated with sampling methods, *ii*) approximation associated with variational Bayes approach [6], and *iii*) computational complexity associated with hyperparameter tuning [110].

2. We provided an analytical evaluation of the sufficient conditions for exact recovery of the number of topics in topic models. Moreover, we provided a bound on the sum of squared errors in terms of the model parameters such as number of documents, vocabulary size, and number of words in each document. We showed that the reconstruction error is $\mathcal{O}(\sqrt{M/n})$, where M/n is the ratio of the number of document to the number of words per document.

3. We provided an accelerated algorithm to solve the proposed convex optimization problem. We reformulate the problem in the dual form. By evaluating the duality

gap, we were able to provide accuracy guarantees for the algorithm. We evaluate our theoretical results on synthetic datasets. Finally, we applied the proposed method on two image datasets and three real world text datasets to illustrate how the method can be applied to perform dimension reduction.

In the continuous domain,

1. We developed a new entropy estimator based on the principle of maximum entropy and greedy m-term approximation. We also provided the analysis of the estimation error, specifically an in-probability error bound in terms of the problem parameters (e.g., number of samples, number of the approximation terms). The error of the proposed estimator is $\mathcal{O}(\sqrt{\log n/n})$; only a factor of $\sqrt{\log n}$ away from the classical statistical parameter estimation error $\mathcal{O}(\sqrt{1/n})$. The application of the method to anomaly detection in sensor networks was demonstrated. Our proposed estimator was shown to be competitive with other approaches.

2. We proposed the maximum entropy framework for entropy estimation. The proposed estimators deploy m-term approximation to estimate the entropy. In addition to a brute-force estimator, we introduced a low computational complexity greedy m-term entropy estimator. Theoretical analysis of the proposed estimators shows that the estimation error is $\mathcal{O}(\sqrt{\log n/n})$. As with other entropy estimation methods, the proposed method can be used for a variety of applications. The application of the method to anomaly detection in sensor networks was demonstrated. Our proposed estimator was shown to be competitive with other approaches.

3. We introduced a new framework for MIL using the principle of maximum entropy approach. A metric defined over the space of the distributions was introduced to measure the similarities among bags in MIL.

4. We proposed confidence-constrained maximum entropy to jointly learn the space of distributions and an accelerated proximal gradient approach was proposed to solve the convex optimization problem.

5. The performance of the proposed approach was evaluated in terms of rank recovery in the space of distributions and compared with regularized MaxEnt. We examined the classification accuracy of CCMaxEnt on four real world dataset and compared the results with the state-of-the-art algorithms in MIL.

6.2 Publications

In this part, a list of our publications which were written during the course of the Ph.D. is provided.

1. Behmardi, B., Briggs, F., Fern, X., and Raich R. *Confidence-Constrained Maximum Entropy Framework for Learning Multi-instance data*, IEEE Transactions on Signal Processing, submitted, 2012.

2. Behmardi, B. and Raich, R. *On confidence-constrained rank recovery in topic models*, IEEE Transactions on Signal Processing, Volume: 60, Issue: 10, page(s): 5146–5162 [14].

3. Behmardi, B., Briggs, F., Fern, X., and Raich R. *Regularized joint density estimation for multi-instance learning*, In Proceedings of IEEE International Workshop on Statistical Signal Processing, page(s): 740-743, 2012 [9].

4. Behmardi, B. and Raich, R. *Convex optimization for exact rank recovery in topic models*, In Proceedings of IEEE International Workshop on Machine Learning for

Signal Processing, page(s): 1-6, 2011 [12].

5. Behmardi, B. and Raich, R. *On provable exact low-rank recovery in topic models*, In Proceedings of IEEE International Workshop on Statistical Signal Processing, page(s): 265-268, 2011 [13].

6. Behmardi, B., Raich, R., and Hero, A.O., *Entropy estimation using the principle of maximum entropy*, In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, page(s): 2008-2011, 2011 [15].

7. Behmardi, B. and Raich, R. *Isometric correction for manifold learning*, AAAI symposium on manifold learning, pages: 1-9, 2010 [11].

6.3 Future research

In the following, we present a few directions for future research.

1. **Theoretical proof for nuclear norm minimization:** The rank function in CRM was heuristically replaced with nuclear norm in CNM. Nuclear norm minimization produces a low-rank solution in practice, but a theoretical characterization of when CNM can produce the minimum rank solution was not investigated. The mathematical characterization of minimum rank solution was provided in the case where the constraints were affine [97, 98]. The extension of theoretical guarantees to the nonlinear set of inequalities is an open research direction.

2. **Supervised approach:** Our approach is an unsurprised technique in dimension reduction. Developing a new model which accounts for the useful discriminative information in the dataset is another future research direction.

3. **Exact evaluation of the sufficient conditions:** The sufficient conditions proposed in this dissertation in the discrete domain depends on the distribution of the smallest singular value σ_T of matrix Ψ. In our experiment, we evaluated the probability of $P(\sigma_T \geq 2\epsilon^*)$ empirically. Knowing the distribution of the smallest singular value of matrix Ψ results to the exact computation of $P(\sigma_T \geq 2\epsilon^*)$. Note that the distribution of the smallest singular value is highly dependent to the sampling process used for generating matrix Ψ. This limits the generality of the approach. However, one can consider a special case of the sampling process (e.g., LDA) and develop the bound including the hyperparameters of the LDA sampling process.

Bibliography

[1] L. AlSumait, D. Barbará, J. Gentle, and C. Domeniconi. Topic significance ranking of LDA generative models. *Journal of Machine Learning and Knowledge Discovery in Databases*, pages 67–82, 2009.

[2] S.I. Amari. Differential geometry of curved exponential families-curvatures and information loss. *The Annals of Statistics*, pages 357–385, 1982.

[3] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. *Proceedings of Advances in Neural Information Processing Systems*, 15:561–568, 2002.

[4] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *Proceedings of Conference on Advances in Neural Information Processing Systems*, pages 561–568, 2003.

[5] A. Asuncion, M. Welling, P. Smyth, and Y.W. Teh. On smoothing and inference for topic models. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 27–34. AUAI Press, 2009.

[6] H. Attias. Inferring parameters and structure of latent variable models by variational Bayes. In *Proceedings of 15th Conference on Uncertainty in Artificial Intelligence*, volume 2, 1999.

[7] F.R. Bach. Consistency of trace norm minimization. *Journal of Machine Learning Research*, 9:1019–1048, 2008.

[8] A.R. Barron and C.H. Sheu. Approximation of density functions by sequences of exponential families. *The Annals of Statistics*, 19(3):1347–1369, 1991.

[9] B. Behmardi, F. Briggs, X. Fern, and Raich R. Regularized joint density estimation for multi-instance learning. In *Proceedings of IEEE International Workshop on Statistical Signal Processing*, pages 740–743, 2012.

[10] B. Behmardi and R. Raich. Entropy estimation using the principle of maximum entropy. Technical report, Oregon State University, Department of Electrical Engineering and Computer Science, October 2010.

[11] B. Behmardi and R. Raich. Isometric correction for manifold learning. In *AAAI symposium on manifold*, pages 1–9, 2010.

[12] B. Behmardi and R. Raich. Convex optimization for exact rank recovery in topic models. In *Proceedings of IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6, 2011.

[13] B. Behmardi and R. Raich. On provable exact low-rank recovery in topic models. In *Proceedings of IEEE International Workshop on Statistical Signal Processing*, pages 265 –268, 2011.

[14] B. Behmardi and R. Raich. On confindence-constrained rank recovery in topic models. *IEEE Transactions on Signal Processing*, 60(10):5146–5162, 2012.

[15] B. Behmardi, R. Raich, and A.O. Hero. Entropy estimation using the principle of maximum entropy. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2008–2011. IEEE, 2011.

[16] J. Beirlant, EJ Dudewicz, L. Györfi, and E.C. Van der Meulen. Nonparametric entropy estimation: An overview. *International Journal of Mathematical and Statistical Sciences*, 6:17–40, 1997.

[17] A. Berger. The improved iterative scaling algorithm: A gentle introduction. *Technical Report*, 1997.

[18] A.L. Berger, V.J.D. Pietra, and S.A.D. Pietra. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71, 1996.

[19] D. Bertsekas. On the Goldstein-Levitin-Polyak gradient projection method. *IEEE Transactions on Automatic Control*, 21(2):174–184, 1976.

[20] David M. Blei and Jon D. McAuliffe. Supervised topic models. In *Proceedings of Conference on Advances in Neural Information Processing Systems*, 2007.

[21] D.M. Blei. Introduction to probabilistic topic models. *Available from http://www.cs.princeton.edu/ blei/papers/*, 2011.

[22] D.M. Blei, T.L. Griffiths, and M.I. Jordan. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of the ACM*, 57(2):7, 2010.

[23] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[24] Z.I. Botev, J.F. Grotowski, and D. Kroese. Kernel density estimation via diffusion. *Annals of Statistics*, 38(5), 2010.

[25] S.P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[26] R.C. Bunescu and R.J. Mooney. Multiple instance learning for sparse positive bags. In *Proceedings of International Conference on Machine Learning*, pages 105–112. ACM, 2007.

[27] J.F. Cai, E.J. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion. *Journal on Optimization*, 20:615–640, 2008.

[28] E.J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis. *Journal of ACM*, 58(1):1–37, 2009.

[29] E.J. Candes and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

[30] E.J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[31] K.M. Carter, R. Raich, W.G. Finn, and A.O. Hero. Information preserving component analysis: Data projections for flow cytometry analysis. *Selected Topics in Signal Processing, IEEE Journal of*, 3(1):148–158, 2009.

[32] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:1–27, 2011. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[33] P. Chen and D. Suter. Recovering the missing components in a large noisy low-rank matrix: Application to SFM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1051–1063, 2004.

[34] S.F. Chen and R. Rosenfeld. A survey of smoothing techniques for me models. *IEEE Transactions on Speech and Audio Processing*, 8(1):37–50, 2000.

[35] X. Chen, X. Hu, X. Shen, and G. Rosen. Probabilistic topic modeling for genomic data interpretation. In *IEEE International Conference on Bioinformatics and Biomedicine*, pages 149–152. IEEE, 2010.

[36] Y. Chen and J.Z. Wang. Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research*, 5:913–939, 2004.

[37] I. Csiszár and P.C. Shields. *Information theory and statistics: A tutorial*, volume 1. Communication and information theroy, 2004.

[38] I. Csiszár and P.C. Shields. *Information theory and statistics: A tutorial*, volume 1. Communication and information theroy, 2004.

[39] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[40] S. Della Pietra, V. Della Pietra, and J. Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, 1997.

[41] R.A. DeVore and V.N. Temlyakov. Nonlinear approximation in finite-dimensional spaces. *Journal of Complexity*, 13(4):489–508, 1997.

[42] T.G. Dietterich, R.H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.

[43] T.T. Do, Y. Chen, N. Nguyen, L. Gan, and T.D. Tran. A fast and efficient heuristic nuclear-norm algorithm for affine rank minimization. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3393–3396. IEEE, 2009.

[44] D.L. Donoho, I. Drori, Y. Tsaig, and J.L. Starck. Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit. *Citeseer*, 2006.

[45] M. Dudik, D.M. Blei, and R.E. Schapire. Hierarchical maximum entropy density estimation. In *Proceedings of the 24th international conference on Machine learning*, pages 249–256. ACM, 2007.

[46] M. Dudik, S. Phillips, and R. Schapire. Performance guarantees for regularized maximum entropy density estimation. *Learning Theory*, pages 472–486, 2004.

[47] M. Dudık, S.J. Phillips, and R.E. Schapire. Maximum entropy density estimation with generalized regularization and an application to species distribution modeling. *Journal of Machine Learning Research*, 8:1217–1260, 2007.

[48] P. Duygulu, K. Barnard, J. De Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. *Proceedigs of European Conference on Computer Vision*, pages 349–354, 2006.

[49] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[50] M. Fazel. *Matrix rank minimization with applications*. PhD thesis, Stanford University, 2002.

[51] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *Proceeding of IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 524–531, 2005.

[52] T. Gärtner, P.A. Flach, A. Kowalczyk, and A.J. Smola. Multi-instance kernels. In *Proceedings of International Conference on Machine Learning*, pages 179–186, 2002.

[53] Z. Ghahramani, P. Sollich, and T. L. Griffiths. Bayesian nonparametric latent feature models. *Bayesian Statistics*, 2007.

[54] P.E. Gill, W. Murray, and M.H. Wright. *Practical optimization*, volume 1. Academic press, 1981.

[55] A.A. Goldstein. Convex programming in Hilbert space. *American Mathematics Society*, 70(5):709–710, 1964.

[56] J. Goodman. Exponential priors for maximum entropy models. In *Proc. HLT-NAACL*, pages 305–312, 2004.

[57] T.L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101, 2004.

[58] J.P. Haldar and D. Hernando. Rank-constrained solutions to linear matrix equations using powerfactorization. *Signal Processing Letters*, 16(7):584–587, 2009.

[59] E.T. Hale, W. Yin, and Y. Zhang. A fixed-point continuation method for l1-regularized minimization with applications to compressed sensing. *CAAM TR07-07, Rice University*, 2007.

[60] C. Henegar, K. Clément, and J.D. Zucker. Unsupervised multiple-instance learning for functional profiling of genomic data. *European Conference on Machine Learning*, pages 186–197, 2006.

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

[62] M. Hoffman, D. Blei, and P. Cook. Bayesian nonparametric matrix factorization for recorded music. In *Proceedings of International Conferences on Machine Learning*, pages 439–446, 2010.

[63] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval*, pages 50–57. ACM, 1999.

[64] E.T. Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.

[65] R. Jin, S. Wang, and Z.H. Zhou. Learning a distance metric from multi-instance multi-label data. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 896–902, 2009.

[66] B. Kanagal and V. Sindhwani. Rank selection in low-rank matrix approximations: A study of cross-validation for NMFs. In *Proceedings of Conference on Advances in Neural Information Processing Systems*, volume 1, pages 10–15, 2010.

[67] J. Kazama and J. Tsujii. Evaluation and extension of maximum entropy models with inequality constraints. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 137–144. Association for Computational Linguistics, 2003.

[68] R.H. Keshavan, A. Montanari, and S. Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 99:2057–2078, 2010.

[69] K. Konishi and T. Furukawa. A nuclear norm heuristic approach to fractionally spaced blind channel equalization. *Signal Processing Letters*, 18(1):59–62, 2011.

[70] B. Krishnapuram, L. Carin, M.A.T. Figueiredo, and A.J. Hartemink. Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(6):957–968, 2005.

[71] B. Lakshminarayanan and R. Raich. Inference in supervised latent Dirichlet allocation. In *Proceedings of IEEE International Workshop on Machine Learning for Signal Processing*. IEEE, 2011.

[72] C. Lemaréchal and C. Sagastizábal. Practical aspects of the Moreau-Yosida regularization I: theoretical properties. *Rapport de Recherche-Institut National de Recherche en Informatique et en Automatique*, 1994.

[73] Z. Lin, M. Chen, L. Wu, and Y. Ma. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. *Mathematical Programming*, 2009.

[74] Y.J. Liu, D. Sun, and K.C. Toh. An implementable proximal point algorithmic framework for nuclear norm minimization. *Mathematical Programming*, pages 1–38, 2009.

[75] Z. Liu and L. Vandenberghe. Semidefinite programming methods for system realization and identification. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 4676–4681. IEEE, 2009.

[76] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the 7th IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157. IEEE, 1999.

[77] S. Ma, D. Goldfarb, and L. Chen. Fixed point and bregman iterative methods for matrix rank minimization. *Mathematical Programming*, 128(1):321–353, 2011.

[78] R. Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the sixth conference on natural language learning (CoNLL-2002)*, pages 49–55, 2002.

[79] J.H. Manton, R. Mahony, and Y. Hua. The geometry of weighted low-rank approximations. *IEEE Transactions on Signal Processing*, 51(2):500–514, 2003.

[80] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In *Proceedings of Advances in Neural Information Processing Systems*, pages 570–576, 1998.

[81] O. Maron and A.L. Ratan. Multiple-instance learning for natural scene classification. In *Proceedings of 15th International Conference on Machine Learning*, 1998.

[82] R. Meka, P. Jain, C. Caramanis, and I.S. Dhillon. Rank minimization via online learning. In *Proceedings of the 25th International Conference on Machine learning*, pages 656–663. ACM, 2008.

[83] R. Meka, P. Jain, and I.S. Dhillon. Guaranteed rank minimization via singular value projection. In *Proceedings of Conference on Advances in Neural Information Processing Systems*, 2010.

[84] K. Mohan and M. Fazel. Reweighted nuclear norm minimization with application to system identification. In *Proceedings of Conference on American Control*, pages 2953–2959. IEEE, 2010.

[85] S. Negahban and M.J. Wainwright. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *Submitted to the Annals of Statistics*, 2009.

[86] Y. Nesterov. A method of solving a convex programming problem with convergence rate O (1/k2). *Soviet Mathematics Doklady*, 27:372–376, 1983.

[87] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *IJCAI workshop on machine learning for information filtering*, volume 1, pages 61–67, 1999.

[88] M. Nilsson and W.B. Kleijn. On the estimation of differential entropy from data located on embedded manifolds. *IEEE Transactions on Information Theory*, 53(7):2330–2341, 2007.

[89] J. Paisley, X. Liao, and L. Carin. Active learning and basis selection for kernel-based linear models: a bayesian perspective. *IEEE Transactions on Signal Processing*, 58(5):2686–2700, 2010.

[90] T. Papadopoulo and M. Lourakis. Estimating the jacobian of the singular value decomposition: Theory and applications. *Computer Vision-ECCV 2000*, pages 554–570, 2000.

[91] P.O. Perry and P.J. Wolfe. Minimax rank estimation for subspace tracking. *IEEE Journal of Selected Topics in Signal Processing*, 4(3):504–513, 2010.

[92] S.J. Phillips, M. Dudík, and R.E. Schapire. A maximum entropy approach to species distribution modeling. In *Proceedings of the twenty-first international conference on Machine learning*, page 83. ACM, 2004.

[93] T.K. Pong, P. Tseng, S. Ji, and J. Ye. Trace norm regularization: Reformulations, algorithms, and multi-task learning. *Submitted to SIAM Journal on Optimization*, 2009.

[94] A. Rahimi and B. Recht. Random features for large-scale kernel machines. *Proceedings of Advances in Neural Information Processing Systems*, 20:1177–1184, 2007.

[95] D. Ramage, E. Rosen, J. Chuang, C.D. Manning, and D.A. McFarland. Topic modeling for the social sciences. In *NIPS Workshop on Applications for Topic Models: Text and Beyond*, 2009.

[96] J. Ramon and L. De Raedt. Multi instance neural networks. In *Proceedings of ICML-2000, Workshop on Attribute-Value and Relational Learning*, pages 53–60, 2000.

[97] B. Recht, M. Fazel, and P.A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, 2007. *SIAM Review*, 52:471–501, 2010.

[98] B. Recht, W. Xu, and B. Hassibi. Necessary and sufficient conditions for success of the nuclear norm heuristic for rank minimization. In *Proceedings of 47th IEEE Conference on Decision and Control*, pages 3065–3070. IEEE, 2008.

[99] R. Salakhutdinov, S.T. Roweis, Z. Ghahramani, et al. On the convergence of bound optimization algorithms. In *Uncertainty in Artificial Intelligence*, volume 19, pages 509–516, 2003.

[100] G. Salton and M.J. McGill. *Introduction to modern information retrieval*. McGraw-Hill, 1986.

[101] D.W. Scott. On optimal and data-based histograms. *Biometrika*, 66(3):605, 1979.

[102] J. Skilling and RK Bryan. Maximum entropy image reconstruction-general algorithm. *Monthly Notices of the Royal Astronomical Society*, 211:111, 1984.

[103] N. Srebro, J.D.M. Rennie, and T. Jaakkola. Maximum-margin matrix factorization. In *Proceedings of Conference on Advances in Neural Information Processing Systems*, volume 17, pages 1329–1336, 2005.

[104] G.W. Stewart. On the early history of the singular value decomposition. *SIAM review*, pages 551–566, 1993.

[105] M. Steyvers and T. Griffiths. Probabilistic topic model. *Handbook of Latent Semantic Analysis*, pages 1–15, 2007.

[106] G. Tang and A. Nehorai. Lower bounds on the mean-squared error of low-rank matrix reconstruction. *Signal Processing, IEEE Transactions on*, 59(10):4559–4571, 2011.

[107] Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[108] K.C. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Optimization*, 6:615–640, 2010.

[109] P. Viola, J. Platt, and C. Zhang. Multiple instance boosting for object detection. In *Proceedings of Advances in Neural Information Processing Systems*, volume 18, pages 1417–1426, 2006.

[110] H. Wallach, D. Mimno, and A. McCallum. Rethinking LDA: Why priors matter. In *Proceedings of Conference on Advances in Neural Information Processing Systems*, volume 22, pages 1973–1981, 2009.

[111] H. Wang, H. Huang, F. Kamangar, F. Nie, and C. Ding. Maximum margin multi-instance learning. In *Proceedings of Advances in Neural Information Processing Systems*, volume 15, pages 341–349, 2011.

[112] Jun Wang, Zucker, and Jean-Daniel. Solving multiple-instance problem: A lazy learning approach. In Pat Langley, editor, *Proceedings of International Conference on Machine Learning*, pages 1119–1125, 2000.

[113] M. Wax and T. Kailath. Detection of signals by information theoretic criteria. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 33(2):387–392, 1985.

[114] M. Welling, C. Chemudugunta, and N. Sutter. Deterministic latent variable models and their pitfalls. In *Proceedings of International Conference on Data Mining*, 2008.

[115] X. Xu and E. Frank. Logistic regression and boosting for labeled bags of instances. *Advances in Knowledge Discovery and Data Mining*, pages 272–281, 2004.

[116] Y. Xu, W. Ping, and A.T. Campbell. Multi-instance metric learning. In *Proceedings of IEEE International Conference on Data Mining*, pages 874–883, 2011.

[117] J. Yu, SVN Vishwanathan, S. Günter, and N.N. Schraudolph. A quasi-newton approach to nonsmooth convex optimization. *A. McCallum and S. Roweis, editors*, 951:1216–1223, 2008.

[118] Z.J. Zha, X.S. Hua, T. Mei, J. Wang, G.J. Qi, and Z. Wang. Joint multi-label multi-instance learning for image classification. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[119] D. Zhang, F. Wang, L. Si, and T. Li. M 3 ic: maximum margin multiple instance clustering. In *Proceedings of International Joint Conferences on Artificial Intelligence*, volume 9, pages 1339–1344, 2009.

[120] M.L. Zhang and Z.H. Zhou. A k-nearest neighbor based algorithm for multi-label classification. In *Proceedings of IEEE International Conference on Granular Computing*, volume 2, pages 718–721, 2005.

[121] M.L. Zhang and Z.H. Zhou. Multi-instance clustering with applications to multi-instance prediction. *Applied Intelligence*, 31(1):47–68, 2009.

[122] Q. Zhang and S.A. Goldman. Em-dd: An improved multiple-instance learning technique. In *Proceedings of Advances in Neural Information Processing Systems*, volume 14, pages 1073–1080. Cambridge, MA: MIT Press, 2001.

[123] Q. Zhang, S.A. Goldman, W. Yu, and J.E. Fritts. Content-based image retrieval using multiple-instance learning. In *Proceedings of International Workshop on Machine Learning*, pages 682–689, 2002.

[124] Z.H. Zhou, K. Jiang, and M. Li. Multi-instance learning based web mining. *Applied Intelligence*, 22(2):135–147, 2005.

[125] S. Zhu, X. Ji, W. Xu, and Y. Gong. Multi-labelled classification using maximum entropy method. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 274–281. ACM, 2005.