

# LEARNING WITHOUT LABELS AND NONNEGATIVE TENSOR FACTORIZATION

A Thesis  
Presented to  
The Academic Faculty

by

Krishnakumar Balasubramanian

In Partial Fulfillment  
of the Requirements for the Degree  
Master of Science in the  
College of Computing

Georgia Institute of Technology  
March 2010

# LEARNING WITHOUT LABELS AND NONNEGATIVE TENSOR FACTORIZATION

Approved by:

Professor Guy Lebanon  
College of Computing  
*Georgia Institute of Technology*

Professor Haesun Park  
College of Computing  
*Georgia Institute of Technology*

Professor Alexander Gray  
College of Computing  
*Georgia Institute of Technology*

Date Approved: 31 March 2010

## ACKNOWLEDGEMENTS

I would like to thank my advisors Prof. Guy Lebanon and Prof. Haesun Park for providing me with opportunities to work in both theoretical and algorithmic/practical aspects of machine learning. Prof. Lebanon taught me to think with a stage-dependent optimal bandwidth during different stages of research, from identifying a problem, to solving it, to conveying it clearly to the community. Prof. Park always guided me in the right directions in research in spite of her busy schedule.

I thank Prof. Alexander Gray for agreeing to be on my thesis committee. I was introduced to several concepts in machine learning through his course. I sincerely thank my co-authors Pinar Donmez, Joshua Dillon, Yi Mao and Jingu Kim. I would also like to thank all my lab mates in 1305 for all the fun.

Special thanks to my parents and friends for their love and support.

# TABLE OF CONTENTS

DEDICATION . . . . .	iii
ACKNOWLEDGEMENTS . . . . .	iv
SUMMARY . . . . .	xiii
I INTRODUCTION . . . . .	1
1.1 Learning without labels . . . . .	1
1.2 Nonnegative Tensor Factorization . . . . .	3
1.2.1 Modeling with tensors . . . . .	5
II UNSUPERVISED ERROR RATE ESTIMATION OF PREDICTORS . . . . .	7
2.1 Definitions . . . . .	7
2.2 Unsupervised Risk Estimation Framework . . . . .	9
2.2.1 Non-Collaborative Estimation of the Risks . . . . .	11
2.2.2 Classification . . . . .	11
2.2.3 Regression . . . . .	12
2.2.4 Noisy Gaussian Channel . . . . .	14
2.2.5 Collaborative Estimation of the Risks: Conditionally Independent Predictors . . . . .	14
2.2.6 Collaborative Estimation of the Risks: Conditionally Correlated Predictors . . . . .	17

2.3	Extensions: Missing Values, Active Learning, and Semi-Supervised Learning . . . . .	18
2.4	Consistency of $\hat{\theta}_n^{\text{mle}}$ and $\hat{R}(f_j)$ . . . . .	20
2.4.1	Consistency of Classification Risk Estimation . . . . .	22
2.4.2	Consistency of Regression Risk Estimation . . . . .	29
2.5	Asymptotic Variance of $\hat{\theta}_n^{\text{mle}}$ and $\hat{R}$ . . . . .	31
2.6	Optimization Algorithms . . . . .	33
2.7	Empirical Evaluation . . . . .	34
III	TRAINING MARGIN BASED CLASSIFIERS WITHOUT LABELS . . . . .	45
3.1	Definitions . . . . .	45
3.2	Unsupervised Risk Estimation . . . . .	47
3.2.1	Asymptotic Normality of $f_\theta(X) Y$ . . . . .	49
3.2.2	Unsupervised Consistency . . . . .	53
3.3	Application 1: Estimating Risk in Transfer Learning . . . . .	55
3.4	Application 2: Unsupervised Learning of Classifiers . . . . .	56
3.4.1	Inaccurate Specification of $p(Y)$ . . . . .	64
IV	ASYMPTOTIC ANALYSIS OF GENERATIVE SEMISUPERVISED LEARNING . . . . .	65
4.1	Overview . . . . .	65
4.2	Related Work . . . . .	66

4.3	Stochastic SSL Estimators . . . . .	67
4.4	A1: Consistency (Classification) . . . . .	69
4.5	A2: Accuracy (Classification) . . . . .	71
4.6	A3: Consistency (Structured) . . . . .	73
4.7	A4: Accuracy (Structured) . . . . .	75
4.8	A5: Tradeoff . . . . .	77
4.9	A6: Practical Algorithms . . . . .	80
V	FAST ALGORITHM FOR NONNEGATIVE TENSOR FACTORIZATION . .	81
5.1	Notations and operations . . . . .	81
5.1.1	Nonnegative tensor factorization . . . . .	82
5.2	NTF using ANLS framework and Block Principal Pivoting . . . . .	83
5.2.1	ANLS framework . . . . .	83
5.2.2	Block Principal Pivoting Algorithm . . . . .	85
5.3	Extensions to Regularized and Sparse NTF . . . . .	88
5.4	Experiments and Results . . . . .	89
5.4.1	Description of data sets . . . . .	90
5.4.2	Timing comparison . . . . .	92
5.4.3	Factor recovery in presence of noise . . . . .	93
5.4.4	Topic identification using NTF . . . . .	94

VI	DISCUSSION . . . . .	99
6.1	Learning without Labels . . . . .	99
6.2	Nonnegative Tensor Factorization . . . . .	100
	REFERENCES . . . . .	101

## LIST OF FIGURES

1	3-way PARAFAC model: The tensor is represented as a linear combination of $r$ rank-1 tensors. This will provide a rank- $r$ approximation to the original tensor. . . . .	5
2	A plot of the loglikelihood functions $\ell(\theta)$ in the case of classification for $k = 1$ (left, $\theta^{\text{true}} = 0.75$ ) and $k = 2$ (right, $\theta^{\text{true}} = (0.8, 0.6)^\top$ ). . . . .	16
3	A plot of the loglikelihood function $\ell(\theta)$ in the case of regression for $k = 1$ with $\theta^{\text{true}} = 0.3$ , $\tau = 1$ , $\mu_y = 0$ and $\sigma_y = 0.2$ . . . . .	16
4	Average value of $ \hat{\theta}_n^{\text{mle}} - \theta^{\text{true}} $ as a function of $\theta^{\text{true}}$ and $p(y = 1)$ for $k = 1$ classifier and $n = 500$ . . . . .	35
5	Scatter plot contrasting the true and predicted values of $\theta$ in the case of a single classifier $k = 1$ , $p(y = 1) = 0.8$ , and $n = 500$ unlabeled examples. . . . .	36
6	Scatter plot contrasting the true and predicted values of $\theta$ in the case of a single regression model $k = 1$ , $\sigma_y = 1$ , and $n = 1000$ unlabeled examples. . . . .	36
7	Comparison of collaborative and non-collaborative estimation for $k = 10$ classifiers. . . . .	37
8	Comparison of supervised and unsupervised estimation for different values of classifiers with $k = 1, 3, 5, 10$ . . . . .	38
9	A figure comparing the estimator accuracy assuming that the marginal $p(y)$ is misspecified. . . . .	39
10	Mean prediction accuracy for the unsupervised predictor combination scheme in (12) for synthetic data. . . . .	40

11	mae( $\hat{\theta}^{\text{mle}}, \theta^{\text{true}}$ ) as a function of $n$ for different number of annotators $k$ on RTE (left) and TEMP (right) datasets. . . . .	40
12	mae( $\theta^{\text{true}}, \hat{\theta}^{\text{mle}}$ ) as a function of the test set size on the Ringnorm dataset. . . . .	43
13	mae( $\hat{\theta}^{\text{mle}}, \theta^{\text{true}}$ ) for the domain adaptation ( $n = 1000, p(y = 1) = 0.75$ ) and 20 newsgroup ( $n = 15,000, p(y = 1) = 0.05$ for each one-vs-all data). . . . .	44
14	Normality of $f_{\theta}(X) Y = 1$ for different data sets and using different classifiers . . . . .	50
15	Error in estimating logloss for logistic regression classifiers trained on one 20-newsgroup classification task and tested on another. . . . .	56
16	Estimation accuracy of classifiers learned by minimizing the unsupervised logloss estimate $\hat{R}_n$ (72) on RCV1 data using gradient descent. . . . .	59
17	Estimation accuracy of classifiers learned by minimizing the unsupervised logloss estimate $\hat{R}_n$ (72) on RCV1 data using grid search. . . . .	60
18	Estimation accuracy of classifiers learned by minimizing the unsupervised logloss estimate $\hat{R}_n$ (72) on MNIST data using gradient descent. . . . .	61
19	Estimation accuracy of classifiers learned by minimizing the unsupervised logloss estimate $\hat{R}_n$ (72) on MNIST data using grid search. . . . .	62
20	Dependence of $ \hat{R}_n - R_n /R_n$ for logloss (based on synthetic data) on the number of unlabeled examples $n$ and how it changes with the classifier accuracy (acc) and the label marginal $p(Y)$ . . . . .	63
21	Three error measures for the multinomial naive Bayes SSL classifier applied to Reuters RCV1 text data. . . . .	77

22	Log-trace of the theoretical variance and demonstration of the practical applicability of utilizing asymptotic analysis . . . . .	78
23	Original images (1 <sup>st</sup> row) and recovered images using <b>BPP</b> (2 <sup>nd</sup> row), <b>AB</b> (3 <sup>rd</sup> row), and <b>MU</b> (4 <sup>th</sup> row) algorithms when they were computed for the same amount of time. . . . .	95
24	Timing-vs-RSSR graphs. The rows represent data sets: Enron 3-way (1 <sup>st</sup> row), Enron 4-way (2 <sup>nd</sup> row), VAST (3 <sup>rd</sup> row), CMU facial (4 <sup>th</sup> row), and synthetic (5 <sup>th</sup> row) tensors. The columns represent reduced ranks: $r = 10$ (left), $r = 50$ (middle) and $r = 90$ (right). . . . .	97

## LIST OF TABLES

1	Data sets used, their sizes, and amount of speed-up: The values in “Speed up” column represent the amount of average speedup the <b>BPP</b> algorithm has over the second fastest algorithm. . . . .	90
2	Timing comparison on amino acid data set: $\mathbf{X} \in \mathbb{R}_+^{5 \times 201 \times 61}$ . The <i>RSSR</i> values were achieved by several algorithms at the corresponding amount of time. . . . .	95
3	Timing comparison on a synthetic tensor $\mathbf{X} \in \mathbb{R}_+^{173 \times 234 \times 854}$ . Parameters used: $\alpha = 0.4, \beta = 0.2$ , and $\gamma = 0.06$ for regularized NTF, and $\alpha = 0.5, \beta = 0.04$ , and $\gamma = 0.2$ for sparse NTF. . . . .	96
4	Factor recovery under noise on two synthetic tensors in $\mathbb{R}_+^{100 \times 221 \times 631}$ with Gaussian noise with two different variances. A smaller the <i>RSSR</i> value means better the recovery. . . . .	96
5	List of 10 topics identified by sparse NTF along with the percentage of people involved in the communication on each topic. . . . .	96
6	List of 9 topics identified by NTF from the VAST 2007 contest data set. . .	98

## SUMMARY

Supervised learning tasks like building a classifier, estimating the error rate of the predictors, are typically performed with labeled data. In most cases, obtaining labeled data is costly as it requires manual labeling. On the other hand, unlabeled data is available in abundance. In this thesis, we discuss methods to perform supervised learning tasks with no labeled data. We prove consistency of the proposed methods and demonstrate its applicability with synthetic and real world experiments. In some cases, small quantities of labeled data maybe easily available and supplemented with large quantities of unlabeled data (semi-supervised learning). We derive the asymptotic efficiency of generative models for semi-supervised learning and quantify the effect of labeled and unlabeled data on the quality of the estimate. Another independent track of the thesis is efficient computational methods for nonnegative tensor factorization (NTF). NTF provides the user with rich modeling capabilities but it comes with an added computational cost. We provide a fast algorithm for performing NTF using a modified active set method called block principle pivoting method and demonstrate its applicability to social network analysis and text mining.

# CHAPTER I

## INTRODUCTION

### *1.1 Learning without labels*

A common task in machine learning is predicting a response variable  $y \in \mathcal{Y}$  based on an explanatory variable  $x \in \mathcal{X}$ . Assuming a joint distribution  $p(x, y)$  and a loss function  $L(y, \hat{y})$ , a predictor  $f : \mathcal{X} \rightarrow \mathcal{Y}$  is characterized by an expected loss or risk function

$$R(f_\theta(X)) = \mathbf{E}_{p(x,y)}\{L(y, f_\theta(x))\}. \quad (1)$$

Typically, this task is performed with labeled data. But in several cases, labeled data maybe costly to obtain. In some cases, it may be unavailable due to privacy considerations where the predictors are constructed by organizations using training sets with private labels. For example, in medical diagnosis prediction, the predictors  $f_1, \dots, f_k$  may be obtained by  $k$  different hospitals, each using a private internal labeled set. Following the training stage, each hospital releases its predictor to the public who then proceed to estimate  $R(f_1), \dots, R(f_k)$  using a separate unlabeled dataset. In such cases, there is a need for developing ways to predict error rates with only unlabeled data.

Another motivation for using unlabeled data is domain adaptation where predictors that are trained on one domain, are used to predict data from a new domain from which we have only unlabeled data. For example, predictors are often trained on labeled examples drawn from the past but are used at test time to predict data drawn from a new distribution associated with the present. Here the labeled data used to train the predictors will not provide an accurate estimate due to differences in the test and train distributions. Another motivation is companies releasing predictors to clients as black boxes (without their training data) in order to protect their intellectual property. This is the situation in business analytics and consulting. In any case, it is remarkable that without labels we can still accurately estimate

supervised risks.

Also, many popular linear classifiers, such as logistic regression, boosting, or SVM, are trained by optimizing a margin-based risk function. For standard linear classifiers  $\hat{Y} = \text{sign} \sum \theta_j X_j$  with  $Y \in \{-1, +1\}$ ,  $X \in \mathbb{R}^d$  the margin is defined as

$$Y f_\theta(X) \quad \text{where} \quad f_\theta(X) \stackrel{\text{def}}{=} \sum_{j=1}^d \theta_j X_j. \quad (2)$$

Training such classifiers involves choosing a particular value of  $\theta$ . This is done by minimizing the risk or expected loss. Since the risk (dropping  $f$  from the previous notation)  $R(\theta)$  depends on the unknown distribution  $p$ , it is usually replaced during training with its empirical counterpart based on a labeled training set

$$(X^{(1)}, Y^{(1)}), \dots, (X^{(n)}, Y^{(n)}) \stackrel{\text{iid}}{\sim} p \quad (3)$$

leading to the following estimator

$$\hat{\theta}_n = \arg \min_{\theta} R_n(\theta) \quad \text{where}$$

$$R_n(\theta) = \mathbf{E}_{\tilde{p}(X, Y)} L(Y, f_\theta(X)) \quad (4)$$

$$= \frac{1}{n} \sum_{i=1}^n L(Y^{(i)}, f_\theta(X^{(i)})). \quad (5)$$

Note, however, that evaluating and minimizing  $R_n$  requires labeled data (64). While suitable in some cases, in cases, where labeled data is difficult or impossible to obtain, there is need for training classifiers with only unlabeled data.

In some case, a small amount of labeled data maybe available easily and supplemented with large quantities of unlabeled data. It is referred as semi-supervised learning in the literature. It is particularly useful when the costs of obtaining labeled and unlabeled samples are different. In particular, assuming that unlabeled data is more easily available, SSL provides improved modeling accuracy by adding a large number of unlabeled samples to a relatively small labeled dataset. Of particular importance is the dependency of that improvement on the amount of unlabeled and labeled data. In the case of structured prediction

the accuracy of the SSL estimator depends also on the specific manner in which sequences are labeled. The quantitative dependency of quality of estimate on the amount of labeled and unlabeled data helps one to trade-off quality and cost of obtaining labeled samples.

In this thesis we develop methods for the above mentioned classification and error rate estimation tasks with only unlabeled data and mild assumptions. Chapter 2 discusses methods for estimating the error rates of predictors with unlabeled data. It also talks about semi-supervised extensions for the same. Chapter 3 demonstrates a method to build linear margin-based classifiers without any labeled data. In chapter 4, we quantify asymptotic variance of semisupervisedly estimated generative statistical models as a function of number of labeled and unlabeled sample.

Chapters 2 and 3 are joint work with Pinar Donmez and Guy Lebanon. My contributions were Sections 2.2, 2.4.1, 3.2, 3.4 (grid search) and 3.4.1. Chapter 4 is joint work with Joshua Dillon and Guy Lebanon. My contributions were Sections 4.3, 4.4, 4.5 and 4.9.

## ***1.2 Nonnegative Tensor Factorization***

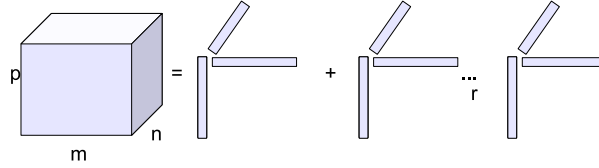
Tensors, which refers to multi-dimensional arrays, provide a mathematical and algorithmic framework for analyzing multi-scale, multi-dimensional data and extracting meaningful information from them. Tensor factorizations [23] are multi-linear generalizations of matrix factorizations and are powerful tools for analyzing multi-dimensional data. Tensor modeling and tensor factorizations enable the analysis of complex data objects, which are not well understood by conventional matrix-based methods. Numerous data sets from various areas have been studied using tensor factorizations [31].

Recently, nonnegative tensor factorization (NTF) [44, 58] has attracted growing interest since it provides interpretable factorizations [45] when data components are inherently non-negative. For example, pixels in digital images, chemical concentrations in chemometrics, high dimensional data in Internet traffic or large-scale social networks [53] are naturally

represented by nonnegative numbers. Once nonnegative factors are obtained, each object can be understood as an additive linear combination of intrinsic parts of the data. This is one of the most important properties of nonnegative matrix factorization (NMF) [34, 35]. When we deal with tensors, a simple extension of the singular value decomposition (SVD) to a higher-order case [33] can be used, but it might fail to factorize in a meaningful way since it does not consider the intrinsic nonnegativity in the data. Instead, NTF has been successfully used to analyze such data sets. An interesting connection between NTF and latent class models in statistics has been also shown [44].

While NTF provides several advantages, modern data sets which tend to be extremely large make the problem computationally challenging. Previous approaches for computing NTF include the following. Andersson and Bro proposed an NTF algorithm based on alternating least squares framework [1], and Kim et. al. proposed an algorithm based on the active set method for the nonnegativity constrained least squares problems [28]. Welling and Webber proposed an algorithm based on multiplicative updates [58]. Friedlander and Hatz proposed an algorithm based on solving bound constrained linear least-squares problem [21]. Despite the presence of such algorithms, there exists need for developing computational faster algorithms, especially for analyzing large data sets.

We propose a fast algorithm for NTF based on the alternating nonnegativity constrained least squares (ANLS) framework [27, 28], in which a series of nonnegativity constrained least squares (NNLS) problems are solved in each iteration. We propose to solve the NNLS problems by a fast active-set-type algorithm, called the block principal pivoting method [43, 29], which overcomes some limitations of the standard active set method. This algorithm is carefully tuned for the special characteristics of the NTF computation and is well suited for large scale data. We also extend our NTF algorithm to other NTF formulations such as regularized NTF and sparse NTF. Comparisons of algorithms using various data sets show that the proposed new algorithm outperforms existing ones in terms of computational speed and approximation accuracy.



**Figure 1:** 3-way PARAFAC model: The tensor is represented as a linear combination of  $r$  rank-1 tensors. This will provide a rank- $r$  approximation to the original tensor.

### 1.2.1 Modeling with tensors

Multi-linear algebra, the algebra of tensors, provides a flexible framework for modeling and analyzing multi-dimensional data. For example, a higher order extension of well-known Eigenfaces approach, called *Tensorfaces* [57], has been proposed for modeling faces for recognition. In Tensorfaces, a set of  $K$  images with each image of pixel size  $N \times N$  is modeled as a  $N \times N \times K$  tensor, rather than  $N^2 \times K$  matrix, and a tensor extension of SVD is calculated for finding the low rank approximation of the data. Tensorfaces provide several advantages over the conventional Eigenfaces method. In web-link analysis, an extension of the popular HITS model was proposed [30] using tensors. The tensor model used here incorporates anchor text information in the adjacency matrix. Given a set of  $I$  web sites and  $K$  keywords, a  $I \times I \times K$  tensor is constructed such that  $x_{ijk}$  is 1 if page  $i$  points to page  $j$  using term  $k$ , and 0 otherwise. Several semantic information may be incorporated by increasing the number of dimensions to more than 3. In both of the examples, the entries of the tensor are nonnegative and incorporating that constraint for factorization will provide a more meaningful lower-rank approximation of data.

Sun et. al., [53] used Tucker decomposition, another type of tensor factorization, for analysis and visualization of social networks. A tensor is formed by taking into account both the network and content aspect of the social network. Then, clustering is performed on the factors of the decomposed tensor to find out which topics people discuss, who are the experts in a given topic, etc. They also provide an efficient method for hierarchical

context-specific visualization of such social network based on tensor decomposition. They demonstrate efficiency of their approach through experiments on social networks.

Chapter 5 is joint work with Haesun Park. Section 5.2.2 is joint work with Jingu Kim.

## CHAPTER II

### UNSUPERVISED ERROR RATE ESTIMATION OF PREDICTORS

#### 2.1 Definitions

A common task in machine learning is predicting a response variable  $y \in \mathcal{Y}$  based on an explanatory variable  $x \in \mathcal{X}$ . Assuming a joint distribution  $p(x, y)$  and a loss function  $L(y, \hat{y})$ , a predictor  $f : \mathcal{X} \rightarrow \mathcal{Y}$  is characterized by an expected loss or risk function

$$R(f_\theta(X)) = \mathbf{E}_{p(x,y)}\{L(y, f_\theta(X))\}. \quad (6)$$

For example, in classification we may have  $\mathcal{X} = \mathbb{R}^d$ ,  $\mathcal{Y} = \{1, \dots, l\}$ , and  $L(y, \hat{y}) = I(y \neq \hat{y})$  where  $I(A) = 1$  if  $A$  is true and 0 otherwise. The resulting risk is known as the 0-1 risk or simply the classification error rate

$$R(f_\theta(X)) = P(f \text{ predicts the wrong class}). \quad (7)$$

In regression we may have  $\mathcal{X} = \mathcal{Y} = \mathbb{R}$ , and  $L(y, \hat{y}) = (y - \hat{y})^2$ . The resulting risk is the mean squared error

$$R(f_\theta(X)) = \mathbf{E}_{p(x,y)}(y - f_\theta(X))^2. \quad (8)$$

For simplicity, we drop  $\theta$  from the notation and use  $R(f)$  to denote the risk of a predictor. We consider the case where we are provided with  $k$  predictors  $f_i : \mathcal{X} \rightarrow \mathcal{Y}$ ,  $i = 1, \dots, k$  ( $k \geq 1$ ) whose risks are unknown. The main task we are faced with is estimating the risks  $R(f_1), \dots, R(f_k)$  without using any labeled data whatsoever. The estimation of  $R(f_i)$  is rather based on an estimator  $\hat{R}(f_i)$  that uses unlabeled data  $x^{(1)}, \dots, x^{(n)} \stackrel{\text{iid}}{\sim} p(x)$ .

A secondary task that we consider is obtaining effective schemes for combining  $k$  predictors  $f_1, \dots, f_k$  in a completely unsupervised manner. We refer to these two tasks of risk

estimation and predictor combination as unsupervised-supervised learning since they refer to unsupervised analysis of supervised prediction models.

It may seem surprising that unsupervised risk estimation is possible at all. After all in the absence of labels there is no ground truth that guides us in estimating the risks. However, if the marginal  $p(y)$  is known it is possible in some cases to obtain a consistent estimator for the risks using only unlabeled data i.e.,

$$\lim_{n \rightarrow \infty} \hat{R}(f_i; x^{(1)}, \dots, x^{(n)}) = R(f_i) \quad \text{with probability 1, } i = 1, \dots, k.$$

In addition to demonstrating consistency, we explore the asymptotic variance of the risk estimators and how it is impacted by changes in  $n$  (amount of unlabeled data),  $k$  (number of predictors), and  $R(f_1), \dots, R(f_k)$  (risks). We also demonstrate that the proposed estimation technique works well in practice on both synthetic and real world data.

The assumption that  $p(y)$  is known seems restrictive, but there are plenty of cases where it holds. Examples include medical diagnosis ( $p(y)$  is the well known marginal disease frequency), handwriting recognition/OCR ( $p(y)$  is the easily computable marginal frequencies of different English letters), regression model for life expectancy ( $p(y)$  is the well known marginal life expectancy tables). In these and other examples  $p(y)$  is obtained from extremely accurate histograms.

The collaborative nature of this diagnosis is especially useful for multiple predictors as the predictor ensemble  $\{f_1, \dots, f_k\}$  diagnoses itself. However, our framework is not restricted to a large  $k$  and works even for a single predictor with  $k = 1$ . It may further be extended to the case of active learning where classifiers are queried for specific data and the case of semi-supervised learning where a small amount of labeled data is augmented by massive unlabeled data.

We proceed in the next section to describe the general framework and some important special cases. In Section 2.3 we discuss extensions to the general framework and in Section 3.2.2-2.5 we discuss the theory underlying our estimation process. In Section 2.6 we discuss practical optimization algorithms. Section 2.7 contains an experimental study.

## 2.2 Unsupervised Risk Estimation Framework

We adopt the framework presented in Section 3.1 with the added requirement that the predictors  $f_1, \dots, f_k$  are stochastic i.e. their prediction  $\hat{y} = f_i(x)$  (conditioned on  $x$ ) is a random variable. Such stochasticity occurs if the predictors are conditional models predicting values according to their estimated probability i.e.,  $f_i$  models a conditional distribution  $q_i$  and predicts  $y'$  with probability  $q_i(y'|x)$ .

As mentioned previously our goal is to estimate the risk associated with classification or regression models  $f_1, \dots, f_k$  based on unlabeled data  $x^{(1)}, \dots, x^{(n)} \stackrel{\text{iid}}{\sim} p(x)$ . The testing marginal and conditional distributions  $p(x), p(y|x)$  may differ from the distributions used at training time for the different predictors. In fact, each predictor may have been trained on a completely different training distribution, or may have been designed by hand with no training data whatsoever. We consider the predictors as black boxes and do not assume any knowledge of their modeling assumptions or training processes.

At the center of our framework is the idea to define a parameter vector  $\theta \in \Theta$  which characterizes the risks  $R(f_1), \dots, R(f_k)$  i.e.  $R(f_j) = g_j(\theta)$  for some function  $g_j : \Theta \rightarrow \mathbb{R}$ ,  $j = 1, \dots, k$ . The parameter vector  $\theta$  is estimated from data by connecting it to the probabilities

$$p_j(y'|y) \stackrel{\text{def}}{=} p(f_j \text{ predicts } y' | \text{ true label is } y).$$

More specifically, we use a plug-in estimate  $\hat{R}(f_j) = g_j(\hat{\theta})$  where  $\hat{\theta}$  maximizes the likelihood of the predictor outputs  $\hat{y}_j^{(i)} = f_j(x^{(i)})$  with respect to the model  $p_\theta(\hat{y}) = \int p_\theta(\hat{y}|y)p(y) dy$ .

The precise equations are:

$$\hat{R}(f_j; \hat{y}^{(1)}, \dots, \hat{y}^{(n)}) = g_j(\hat{\theta}^{\text{mle}}(\hat{y}^{(1)}, \dots, \hat{y}^{(n)})) \quad \text{where} \quad (9)$$

$$\hat{y}^{(i)} \stackrel{\text{def}}{=} (\hat{y}_1^{(i)}, \dots, \hat{y}_k^{(i)})$$

$$\hat{y}_j^{(i)} \stackrel{\text{def}}{=} f_j(x^{(i)})$$

$$\hat{\theta}^{\text{mle}}(\hat{y}^{(1)}, \dots, \hat{y}^{(n)}) = \arg \max \ell(\theta; \hat{y}^{(1)}, \dots, \hat{y}^{(n)}) \quad (10)$$

$$\ell(\theta; \hat{y}^{(1)}, \dots, \hat{y}^{(n)}) = \sum_{i=1}^n \log p_{\theta}(\hat{y}_1^{(i)}, \dots, \hat{y}_k^{(i)}) \quad (11)$$

$$= \sum_{i=1}^n \log \int_{\mathcal{Y}} p_{\theta}(\hat{y}_1^{(i)}, \dots, \hat{y}_k^{(i)} | y^{(i)}) p(y^{(i)}) d\mu(y^{(i)}).$$

The integral in (11) is over the unobserved label  $y^{(i)}$  associated with  $x^{(i)}$ . It should be a continuous integral  $\int_{y^{(i)}=-\infty}^{\infty}$  for regression and a finite summation  $\sum_{y^{(i)}=1}^l$  for classification. For notational simplicity we maintain the integral sign for both cases with the understanding that it is over a continuous or discrete measure  $\mu$ , depending on the topology of  $\mathcal{Y}$ . Note that (11) and its maximizer are computable without any labeled data. All that is required are the classifiers (as black boxes), unlabeled data  $x^{(1)}, \dots, x^{(n)}$ , and the marginal label distribution  $p(y)$ .

Besides being a diagnostic tool for the predictor accuracy,  $\hat{\theta}^{\text{mle}}$  can be used to effectively aggregate  $f_1, \dots, f_j$  to predict the label of a new example  $x^{\text{new}}$

$$\begin{aligned} \hat{y}^{\text{new}} &= \arg \max_{y \in \mathcal{Y}} p_{\hat{\theta}^{\text{mle}}}(y | f_1(x^{\text{new}}), \dots, f_k(x^{\text{new}})) \\ &= \arg \max_{y \in \mathcal{Y}} p(y) \prod_{j=1}^k p_{\hat{\theta}_j^{\text{mle}}}(f_j(x^{\text{new}}) | y). \end{aligned} \quad (12)$$

As a result, our framework may be used to combine existing classifiers or regression models in a completely unsupervised manner.

There are three important research questions concerning the above framework. First, what are the statistical properties of  $\hat{\theta}^{\text{mle}}$  and  $\hat{R}$  (consistency, asymptotic variance). Second, how can we efficiently solve the maximization problem (79). And third, how does the

framework work in practice. We address these three questions in Sections 3.2.2-2.5, 2.6, 2.7 respectively, We devote the rest of the current section to examine some important special cases of (79)-(11) and consider some generalizations in the next section.

### 2.2.1 Non-Collaborative Estimation of the Risks

In the non-collaborative case we estimate the risk of each one of the predictors  $f_1, \dots, f_k$  separately. This reduces the problem to that of estimating the risk of a single predictor, which is repeated  $k$  times for each one of the predictors. We thus assume in this subsection the framework (9)-(11) with  $k = 1$  with no loss of generality. For simplicity we denote the single predictor by  $f$  rather than  $f_1$  and denote  $g = g_1$  and  $\hat{y}^{(i)} = \hat{y}_1^{(i)}$ . The corresponding simplified expressions are

$$\hat{R}(f; \hat{y}^{(1)}, \dots, \hat{y}^{(n)}) = g(\hat{\theta}^{\text{mle}}(\hat{y}^{(1)}, \dots, \hat{y}^{(n)})) \quad (13)$$

$$\hat{\theta}^{\text{mle}}(\hat{y}^{(1)}, \dots, \hat{y}^{(n)}) = \arg \max_{\theta} \sum_{i=1}^n \log \int_{\mathcal{Y}} p_{\theta}(\hat{y}^{(i)} | y^{(i)}) p(y^{(i)}) d\mu(y^{(i)}) \quad (14)$$

where  $\hat{y}^{(i)} = f(x^{(i)})$ .

We consider below several important special cases.

### 2.2.2 Classification

Assuming  $l$  labels  $\mathcal{Y} = \{1, \dots, l\}$ , the classifier  $f$  defines a multivariate Bernoulli distribution  $p_{\theta}(\hat{y}|y)$  mapping the true label  $y$  to  $\hat{y}$

$$p_{\theta}(\hat{y}|y) = \theta_{\hat{y},y}. \quad (15)$$

where  $\theta$  is the stochastic confusion matrix or noise model corresponding to the classifier  $f$ .

In this case, the relationship between the risk  $R(f)$  and the parameter  $\theta$  is

$$R(f) = 1 - \sum_{y \in \mathcal{Y}} \theta_{yy} p(y). \quad (16)$$

Equations (15)-(16) may be simplified by assuming a symmetric error distribution [14]

$$p_{\theta}(\hat{y}|y) = \theta^{I(\hat{y}=y)} \left( \frac{1-\theta}{l-1} \right)^{I(\hat{y} \neq y)} \quad (17)$$

$$R(f) = 1 - \theta \quad (18)$$

where  $I$  is the indicator function and  $\theta \in [0, 1]$  is a scalar corresponding to the classifier accuracy. Estimating  $\theta$  by maximizing (14), with (15) or (17) substituting  $p_{\theta}$  completes the risk estimation task.

In the simple binary case  $l = 2$ ,  $\mathcal{Y} = \{1, 2\}$  with the symmetric noise model (17) the loglikelihood

$$\ell(\theta) = \sum_{i=1}^n \log \sum_{y^{(i)}=1}^2 \theta^{I(\hat{y}^{(i)}=y^{(i)})} (1-\theta)^{I(\hat{y}^{(i)} \neq y^{(i)})} p(y^{(i)}). \quad (19)$$

may be shown to have the following closed form maximizer

$$\hat{\theta}^{\text{mle}} = \frac{p(y=1) - m/n}{2p(y=1) - 1}. \quad (20)$$

where  $m \stackrel{\text{def}}{=} |\{i \in \{1, \dots, n\} : \hat{y}^{(i)} = 2\}|$ . The estimator (20) works well in practice and is shown to be a consistent estimator in the next section (i.e., it converges to the true parameter value). In cases where the symmetric noise model (17) does not hold, using (20) to estimate the classification risk may be misleading. For example, in some cases (20) may be negative. In these cases, using the more general model (15) instead of (17) should provide more accurate results. We discuss this further from theoretical and experimental perspectives in Sections 3.2.2-2.5, and 2.7 respectively.

### 2.2.3 Regression

Assuming a regression equation

$$y = ax + \epsilon, \quad \epsilon \sim N(0, \tau^2)$$

and an estimated regression model or predictor  $\hat{y} = a'x$  we have

$$\hat{y} = a'x = a'a^{-1}(y - \epsilon) = \theta y - \theta \epsilon$$

where  $\theta = a'a^{-1}$ . Thus, in the regression case the distribution  $p_\theta(\hat{y}|y)$  and the relationship between the risk and the parameter  $R(f) = g(\theta)$  are

$$p_\theta(\hat{y}|y) = (2\pi\theta^2\tau^2)^{-1/2} \exp\left(-\frac{(\hat{y} - \theta y)^2}{2\theta^2\tau^2}\right) \quad (21)$$

$$R(f|y) = \text{bias}^2(f) + \text{Var}(f) = (1 - \theta)^2 y^2 + \theta^2 \tau^2 \quad (22)$$

$$R(f) = \theta^2 \tau^2 + (1 - \theta)^2 \mathbf{E}_{p(y)}(y^2). \quad (23)$$

Note that we consider regression as a stochastic estimator in that it predicts  $y = a'x + \epsilon$  or  $y|x \sim N(a'x, \tau^2)$ .

Assuming  $p(y) = N(\mu_y, \sigma_y^2)$  (as is often done in regression analysis) we have

$$p_\theta(\hat{y}^{(i)}) = \int_{\mathbb{R}} p_\theta(\hat{y}^{(i)}|y)p(y)dy = (2\pi\theta^2\tau^2 2\pi\sigma_y^2)^{-1/2} \int_{\mathbb{R}} \exp\left(-\frac{(\hat{y} - \theta y)^2}{2\theta^2\tau^2} - \frac{(y - \mu_y)^2}{2\sigma_y^2}\right) dy \quad (24)$$

$$= \frac{1}{\theta \sqrt{2\pi(\tau^2 + \sigma_y^2)}} \exp\left(\frac{(\hat{y}^{(i)})^2}{2\theta^2\tau^2} \left(\frac{\sigma_y^2}{\sigma_y^2 + \tau^2} - 1\right) + \frac{\mu_y^2}{2\sigma_y^2} \left(\frac{\tau^2}{\sigma_y^2 + \tau^2} - 1\right) + \frac{\hat{y}^{(i)}\mu_y}{\theta(\tau^2 + \sigma_y^2)}\right) \quad (25)$$

where we used the following lemma in the last equation.

**Lemma 2.2.1** (e.g., [41]).

$$\int_{-\infty}^{\infty} A e^{-Bx^2+Cx+D} dx = A \sqrt{\frac{\pi}{B}} \exp(C^2/4B + D) \quad (26)$$

where  $A, B, C, D$  are constants that do not depend on  $x$ .

In this case the loglikelihood simplifies to

$$\ell(\theta) = -n \log\left(\theta \sqrt{2\pi(\tau^2 + \sigma_y^2)}\right) - \left(\frac{\sum_{i=1}^n (\hat{y}^{(i)})^2}{2(\tau^2 + \sigma_y^2)}\right) \frac{1}{\theta^2} + \left(\frac{\mu_y \sum_{i=1}^n \hat{y}^{(i)}}{\tau^2 + \sigma_y^2}\right) \frac{1}{\theta} - n \frac{\mu_y^2}{2(\sigma_y^2 + \tau^2)} \quad (27)$$

which can be shown to have the following closed form maximizer

$$\hat{\theta}^{\text{mle}} = -\frac{\mu_y \sum_{i=1}^n \hat{y}^{(i)}}{2n(\tau^2 + \sigma_y^2)} \pm \sqrt{\frac{(\mu_y \sum_{i=1}^n \hat{y}^{(i)})^2}{4n^2(\tau^2 + \sigma_y^2)^2} + \frac{\sum_{i=1}^n (\hat{y}^{(i)})^2}{n(\tau^2 + \sigma_y^2)}} \quad (28)$$

where the two roots correspond to the two cases where  $\theta = a'/a > 0$  and  $\theta = a'/a < 0$ .

### 2.2.4 Noisy Gaussian Channel

In this case our predictor  $f$  corresponds to a noisy channel mapping a real valued signal  $y$  to its noisy version  $\hat{y}$ . The aim is to estimate the mean squared error or noise level  $R(f) = \mathbf{E} \|y - \hat{y}\|^2$ . In this case the distribution  $p_\theta(\hat{y}|y)$  and the relationship between the risk and the parameter  $R(f) = g(\theta)$  are

$$p_\theta(\hat{y}|y) = (2\pi\theta^2)^{-1/2} \exp\left(-\frac{(\hat{y} - y)^2}{2\theta^2}\right) \quad (29)$$

$$R(f|y) = \theta^2 \quad (30)$$

$$R(f) = \theta^2 \mathbf{E}_{p(y)}(y). \quad (31)$$

The loglikelihood and other details in this case are straightforward variations on the linear regression case described above.

As mentioned above, in both classification and regression, estimating the risks for  $k \geq 2$  predictors rather than a single one may proceed by repeating the optimization process described above for each predictor separately. That is  $\hat{R}(f_j) = g_j(\hat{\theta}_j^{\text{mle}})$  where  $\hat{\theta}_1^{\text{mle}}, \dots, \hat{\theta}_k^{\text{mle}}$  are estimated by maximizing  $k$  different loglikelihood functions. In some cases the convergence rate to the true risks can be accelerated by jointly estimating the risks  $R(f_1), \dots, R(f_k)$  in a collaborative fashion. Such collaborative estimation is possible under some assumptions on the statistical dependency between the noise processes defining the  $k$  predictors. We describe below such an assumption followed by a description of more general cases.

### 2.2.5 Collaborative Estimation of the Risks: Conditionally Independent Predictors

We have previously seen how to estimate the risks of  $k$  predictors by separately applying (9) to each predictor. If the predictors are known to be conditionally independent given the true label i.e.  $p_\theta(\hat{y}_1, \dots, \hat{y}_k|y) = \prod_j p_{\theta_j}(\hat{y}_j|y)$  the loglikelihood (11) simplifies to

$$\ell(\theta) = \sum_{i=1}^n \log \int_{\mathcal{Y}} \prod_{j=1}^k p_{\theta_j}(\hat{y}_j^{(i)}|y^{(i)}) p(y^{(i)}) d\mu(y^{(i)}), \quad \text{where } \hat{y}_j^{(i)} = f_j(x^{(i)}) \quad (32)$$

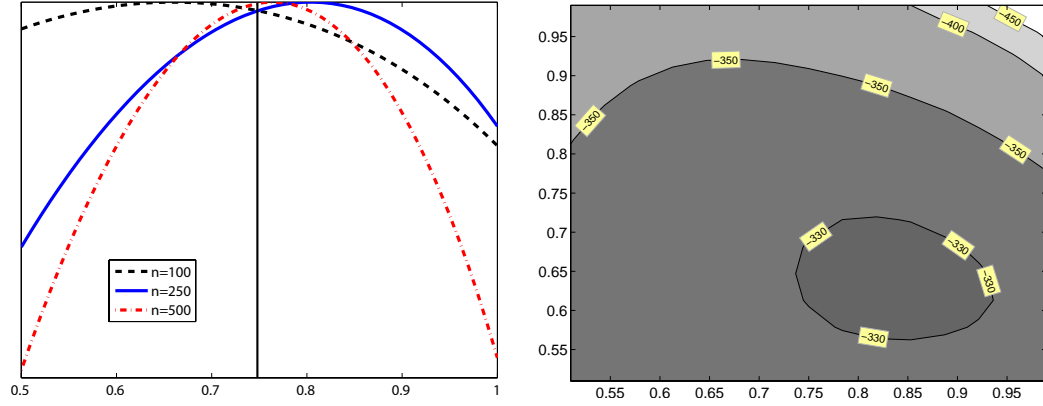
and  $p_{\theta_j}$  above is (15) or (17) for classification and (21) for regression. Maximizing the log-likelihood (32) jointly over  $\theta_1, \dots, \theta_k$  results in estimators  $\hat{R}(f_1), \dots, \hat{R}(f_k)$  that converge to the true value faster than the non-collaborative MLE (14) (more on this in Section 2.7). Equation (32) does not have a closed form maximizer requiring the use of iterative computational techniques.

The conditional independence of the predictors is a much weaker condition than the independence of the predictors which is very unlikely to hold. In our case, each predictor  $f_j$  has its own stochastic noise operator  $T_j(r, s) = p(\hat{y} = r | y = s)$  (regression) or matrix  $[T_j]_{rs} = p_j(\hat{y} = r | y = s)$  (classification) where  $T_1, \dots, T_k$  may be arbitrarily specified. In particular, some predictors may be similar e.g.,  $T_i \approx T_j$ , and some may be different e.g.,  $T_i \not\approx T_j$ . The conditional independence assumption that we make in this subsection is that conditioned on the latent label  $y$  the predictions of the predictors proceed stochastically according to  $T_1, \dots, T_k$  in an independent manner.

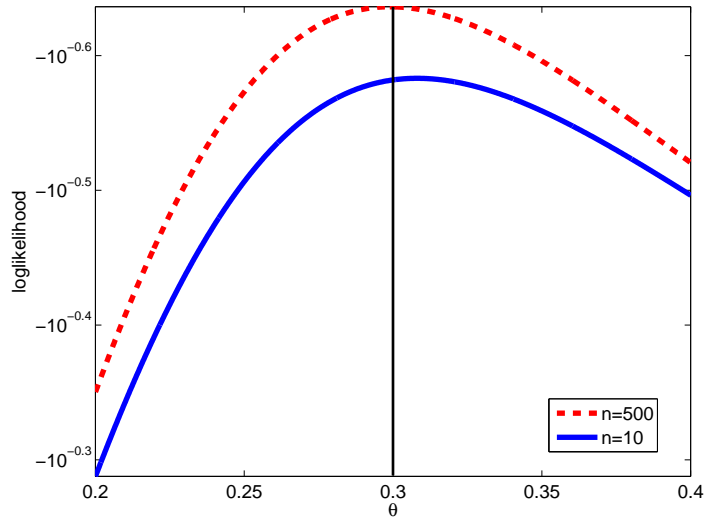
Figure 2 displays the loglikelihood functions  $\ell(\theta)$  for three different dataset sizes  $n = 100, 250, 500$ . As the size  $n$  of the unlabeled data grows the curves become steeper and  $\hat{\theta}_n^{\text{mle}}$  approach  $\theta^{\text{true}}$ . Figure 3 displays a similar figure for  $k = 1$  in the case of regression.

In the case of regression (32) involves an integral over a product of  $k + 1$  Gaussians, assuming that  $y \sim N(\mu_y, \sigma_y^2)$ . In this case the integral in (32) simplifies to  $p_{\theta}(\hat{y}_1^{(i)}, \dots, \hat{y}_k^{(i)})$

$$\begin{aligned}
&= \int_{-\infty}^{\infty} \left( \prod_{j=1}^k \frac{1}{\theta_j \tau \sqrt{2\pi}} e^{-\frac{(\hat{y}_j^{(i)} - \theta_j y^{(i)})^2}{2\theta_j^2 \tau^2}} \right) \frac{1}{\sigma_y \sqrt{2\pi}} e^{-\frac{(y^{(i)} - \mu_y)^2}{2\sigma_y^2}} dy^{(i)} \\
&= \frac{1}{\tau^k (\sqrt{2\pi})^{k+1} \sigma_y \prod_{j=1}^k \theta_j} \int_{-\infty}^{\infty} \exp \left[ -\frac{1}{2} \left( \left( \frac{y^{(i)} - \mu_y}{\sigma_y} \right)^2 + \sum_{j=1}^k \left( \frac{y^{(i)}}{\tau} - \frac{\hat{y}_j^{(i)}}{\tau \theta_j} \right)^2 \right) \right] dy^{(i)} \\
&= \frac{\int_{-\infty}^{\infty} \exp \left( -\frac{1}{2} \left( \frac{1}{\sigma_y^2} + \frac{k}{\tau^2} \right) (y^{(i)})^2 + \left( \frac{\mu_y}{\sigma_y^2} + \sum_{j=1}^k \frac{\hat{y}_j^{(i)}}{\tau^2 \theta_j} \right) y^{(i)} - \frac{1}{2} \left( \frac{\mu_y^2}{\sigma_y^2} + \sum_{j=1}^k \frac{(\hat{y}_j^{(i)})^2}{\tau^2 \theta_j^2} \right) \right) dy^{(i)}}{\tau^k (\sqrt{2\pi})^{k+1} \sigma_y \prod_{j=1}^k \theta_j} \\
&= \frac{\sqrt{\pi} \left[ \frac{1}{2} \left( \frac{1}{\sigma_y^2} + \frac{k}{\tau^2} \right) \right]^{-1/2}}{\tau^k (\sqrt{2\pi})^{k+1} \sigma_y \prod_{j=1}^k \theta_j} \exp \left( \frac{\left( \frac{\mu_y}{\sigma_y^2} + \sum_{j=1}^k \frac{\hat{y}_j^{(i)}}{\tau^2 \theta_j} \right)^2}{2 \left( \frac{1}{\sigma_y^2} + \frac{k}{\tau^2} \right)} - \sum_{j=1}^k \frac{(\hat{y}_j^{(i)})^2}{2\tau^2 \theta_j^2} - \frac{\mu_y^2}{2\sigma_y^2} \right) \quad (33)
\end{aligned}$$



**Figure 2:** A plot of the loglikelihood functions  $\ell(\theta)$  in the case of classification for  $k = 1$  (left,  $\theta^{\text{true}} = 0.75$ ) and  $k = 2$  (right,  $\theta^{\text{true}} = (0.8, 0.6)^\top$ ). The loglikelihood was constructed based on random samples of unlabeled data with sizes  $n = 100, 250, 500$  (left) and  $n = 250$  (right) and  $p(y = 1) = 0.75$ . In the left panel the  $y$  values of the curves were scaled so their maxima would be aligned. For  $k = 1$  the estimators  $\hat{\theta}^{\text{mle}}$  (and their errors  $|\hat{\theta}^{\text{mle}} - 0.75|$ ) for  $n = 100, 250, 500$  are  $0.6633$  ( $0.0867$ ),  $0.8061$  ( $0.0561$ ),  $0.765$  ( $0.0153$ ). As additional unlabeled examples are added the loglikelihood curves become steeper and their maximizers become more accurate and closer to  $\theta^{\text{true}}$ .



**Figure 3:** A plot of the loglikelihood function  $\ell(\theta)$  in the case of regression for  $k = 1$  with  $\theta^{\text{true}} = 0.3$ ,  $\tau = 1$ ,  $\mu_y = 0$  and  $\sigma_y = 0.2$ . As additional unlabeled examples are added the loglikelihood curve become steeper and their maximizers get closer to the true parameter  $\theta^{\text{true}}$  resulting in a more accurate risk estimate.

where the last equation was obtained using Lemma 2.2.1 concerning Gaussian integrals. Note that this equation does not have a closed form maximizer requiring the use of iterative computational techniques.

## 2.2.6 Collaborative Estimation of the Risks: Conditionally Correlated Predictors

In some cases the conditional independence assumption made in the previous subsection does not hold and the factorization (32) is violated. In this section, we discuss how to relax this assumption in the classification case. A similar approach may also be used for regression. We omit the details here due to notational clarity.

There are several ways to relax the conditional independence assumption. Most popular, perhaps, is the mechanism of hierarchical loglinear models for categorical data [8]. For example, generalizing our conditional independence assumption to second-order interaction log-linear models we have

$$\log p(\hat{y}_1, \dots, \hat{y}_k | y) = \alpha_y + \sum_{i=1}^l \beta_{i, \hat{y}_i, y} + \sum_{i < j} \gamma_{i, j, \hat{y}_i, \hat{y}_j, y} \quad (34)$$

where the following ANOVA-type parameter constraints are needed [8]

$$\begin{aligned} 0 &= \sum_{\hat{y}_i} \beta_{i, \hat{y}_i, y} \quad \forall i, y \\ 0 &= \sum_{\hat{y}_i} \gamma_{i, j, \hat{y}_i, \hat{y}_j, y} = \sum_{\hat{y}_j} \gamma_{i, j, \hat{y}_i, \hat{y}_j, y} \quad \forall i, j, y. \end{aligned} \quad (35)$$

The  $\beta$  parameters in (34) correspond to the order-1 interaction between the variables  $\hat{y}_1, \dots, \hat{y}_k$ , conditioned on  $y$ . They correspond to the  $\theta_i$  in the independent formulation (15)-(17). The  $\gamma$  parameters capture two-way interactions which do not appear in the conditionally independent case. Indeed, setting  $\gamma_{i, j, \hat{y}_i, \hat{y}_j, y} = 0$  retrieves the independent models (15)-(17).

In the case of classification, the number of degrees of freedom or free unconstrained parameters in (34) depends on whether the number of classes is 2 or more and what additional assumptions exist on  $\beta$  and  $\gamma$ . For example, assuming that the probability of  $f_i, f_j$

making an error depends on the true class  $y$  but not on the predicted classes  $\hat{y}_i, \hat{y}_j$  results in a  $k + k^2$  parameters. Relaxing that assumption but assuming binary classification results in  $2k + 4k^2$  parameters. The estimation and aggregation techniques described in Section 2.2.5 work as before with a slight modification of replacing (15)-(17) with variations based on (34) and enforcing the constraints (35).

Equation (34) captures two-way interactions but cannot model higher order interactions. However, three-way and higher order interaction models are straightforward generalizations of (34) culminating in the full loglinear model which does not make any assumption on the statistical dependency of the noise operators  $T_1, \dots, T_k$ . However, as we weaken the assumptions underlying the loglinear models and add higher order interactions the number of parameters increases adding to the difficulty in estimating the risks  $R(f_1), \dots, R(f_k)$ .

In our experiments on real world data (see Section 2.7), it is often the case that maximizing the loglikelihood under the conditionally independent assumption (32) provides adequate accuracy and there is no need for the more general (34)-(35). Nevertheless, we include here the case of loglinear models as it may be necessary in some situations.

### ***2.3 Extensions: Missing Values, Active Learning, and Semi-Supervised Learning***

In this section, we discuss extensions to the current framework. Specifically, we consider extending the framework to the cases of missing values, active and semi-supervised learning.

Occasionally, some predictors are unable to provide their output over specific data points. That is assuming a dataset  $x^{(1)}, \dots, x^{(n)}$  each predictor may provide output on an arbitrary subset of the data points  $\{f_j(x^{(i)}) : i \in S_j\}$ , where  $S_j \subset \{1, \dots, n\}, j = 1, \dots, k$ .

Commonly referred to as a missing value situation, this scenario may apply in cases where different parts of the unlabeled data are available to the different predictors at test time due to privacy, computational complexity, or communication cost. Another example where this scenario applies is active learning where operating  $f_j$  involves a certain cost

$c_j \geq 0$  and it is not advantageous to operate all predictors with the same frequency for the purpose of estimating the risks  $R(f_1), \dots, R(f_k)$ . Such is the case when  $f_j$  corresponds to judgments obtained from human experts or expensive machinery that is busy serving multiple clients. Active learning fits into this situation with  $S_j$  denoting the set of selected data points for each predictor.

We proceed in this case by defining indicators  $\beta_{ji}$  denoting whether predictor  $j$  is available to emit  $f_j(x^{(i)})$ . The risk estimation proceeds as before with the observed likelihood modified to account for the missing values.

In the case of collaborative estimation with conditional independence, the estimator and loglikelihood become

$$\begin{aligned} \hat{\theta}_n^{\text{mle}} &= \arg \max_{\theta} \ell(\theta) \\ \ell(\theta) &= \sum_{i=1}^n \log \sum_{r:\beta_{ri}=0} \int_{\mathcal{Y}} p_{\theta}(\hat{y}_1^{(i)}, \dots, \hat{y}_k^{(i)}) d\mu(\hat{y}_r^{(i)}) \\ &= \sum_{i=1}^n \log \sum_{r:\beta_{ri}=0} \iint_{\mathcal{Y}^2} p_{\theta}(\hat{y}_1^{(i)}, \dots, \hat{y}_k^{(i)} | y^{(i)}) p(y^{(i)}) d\mu(\hat{y}_r^{(i)}) d\mu(y^{(i)}) \end{aligned} \quad (36)$$

where  $p_{\theta}$  may be further simplified using the non-collaborative approach, or using the collaborative approach with conditional independence or loglinear model assumptions.

In the case of semi-supervised learning a small set of labeled data is augmented by a large set of unlabeled data. In this case our framework remains as before with the likelihood summing over the observed labeled and unlabeled data. For example, in the case of collaborative estimation with conditional independence we have

$$\ell(\theta) = \sum_{i=1}^n \log \int_{\mathcal{Y}} \prod_{j=1}^k p_{\theta_j}(\hat{y}_j^{(i)} | y^{(i)}) p(y^{(i)}) d\mu(y^{(i)}) + \sum_{i=n+1}^m \log \prod_{j=1}^k p_{\theta_j}(\hat{y}_j^{(i)} | y^{(i)}) p(y^{(i)}). \quad (37)$$

The different variations concerning missing values, active learning, semi-supervised learning, and non-collaborative or collaborative estimation with conditionally independent or correlated noise processes can all be combined in different ways to provide the appropriate likelihood function. This provides substantial modeling flexibility.

## 2.4 Consistency of $\hat{\theta}_n^{\text{mle}}$ and $\hat{R}(f_j)$

In this and the next section we consider the statistical behavior of the estimator  $\hat{\theta}_n^{\text{mle}}$  defined in (79) and the risk estimator  $\hat{R}(f_j) = g_j(\hat{\theta}^{\text{mle}})$  defined in (9). The analysis is conducted under the assumption that the vectors of observed predictors outputs  $\hat{y}^{(i)} = (\hat{y}_1^{(i)}, \dots, \hat{y}_k^{(i)})$  are iid samples from the distribution

$$p_\theta(\hat{y}) = p_\theta(\hat{y}_1, \dots, \hat{y}_k) = \int_{\mathcal{Y}} p_\theta(\hat{y}_1, \dots, \hat{y}_k | y) p(y) d\mu(y).$$

We start by investigating whether estimator  $\hat{\theta}^{\text{mle}}$  in (79) converges to the true parameter value. More formally, strong consistency of the estimator  $\hat{\theta}_n^{\text{mle}} = \hat{\theta}(\hat{y}^{(1)}, \dots, \hat{y}^{(n)})$ ,  $\hat{y}^{(1)}, \dots, \hat{y}^{(n)} \stackrel{\text{iid}}{\sim} p_{\theta_0}$  is defined as strong convergence of the estimator to  $\theta_0$  as  $n \rightarrow \infty$  [20]

$$\lim_{n \rightarrow \infty} \hat{\theta}_n^{\text{mle}}(\hat{y}^{(1)}, \dots, \hat{y}^{(n)}) = \theta_0 \text{ with probability 1.} \quad (38)$$

In other words as the number of samples  $n$  grows, the estimator will surely converge to the true parameter  $\theta_0$  governing the data generation process.

Assuming that the risks  $R(f_j) = g_j(\theta)$  are defined using continuous functions  $g_j$ , strong consistency of  $\hat{\theta}^{\text{mle}}$  implies strong convergence of  $\hat{R}(f_j)$  to  $R(f_j)$ . This is due to the fact that continuity preserves limits. Indeed, as the  $g_j$  functions are continuous in both the classification and regression cases, strong consistency of the risk estimators  $\hat{R}(f_j)$  reduces to strong consistency of the estimators  $\hat{\theta}^{\text{mle}}$ .

It is well known that the maximum likelihood estimator is often strongly consistent. Consider, for example, the following theorem.

**Proposition 1** (e.g., [20]). *Let  $\hat{y}^{(1)}, \dots, \hat{y}^{(n)} \stackrel{\text{iid}}{\sim} p_{\theta_0}$ ,  $\theta_0 \in \Theta$ . If the following conditions hold*

1.  $\Theta$  is compact (compactness)
2.  $p_\theta(\hat{y})$  is upper semi-continuous in  $\theta$  for all  $\hat{y}$  (continuity)
3. There exists a function  $K(\hat{y})$  such that  $\mathbf{E}_{p_{\theta_0}} |K(\hat{y})| < \infty$  (boundedness)  
and  $\log p_\theta(\hat{y}) - \log p_{\theta_0}(\hat{y}) \leq K(\hat{y}) \quad \forall \hat{y} \quad \forall \theta$
4. For all  $\theta$  and sufficiently small  $\rho > 0$ ,  $\sup_{|\theta' - \theta| < \rho} p_{\theta'}(\hat{y})$  is (measurability)  
measurable in  $\hat{y}$
5.  $p_\theta \equiv p_{\theta_0} \Rightarrow \theta = \theta_0$  (identifiability)

then the maximum likelihood estimator is strongly consistent i.e.,  $\hat{\theta}^{\text{mle}} \rightarrow \theta_0$  as  $n \rightarrow \infty$  with probability 1.

Note that  $p_\theta(\hat{y})$  in the proposition above corresponds to  $\int_{\mathcal{Y}} p_\theta(\hat{y}|y)p(y) d\mu(y)$  in our framework. That is the MLE operates on the observed data or predictor output  $\hat{y}^{(1)}, \dots, \hat{y}^{(n)}$  that is sampled iid from the distribution  $p_{\theta_0}(\hat{y}) = \int_{\mathcal{Y}} p_{\theta_0}(\hat{y}|y)p(y) d\mu(y)$ .

Of the five conditions above, the last condition of identifiability is the only one that is truly problematic. The first condition of compactness is trivially satisfied in the case of classification. In the case of regression it is satisfied assuming that the regression parameter and model parameter are finite and  $a \neq 0$  as the estimator  $\hat{\theta}^{\text{mle}}$  will eventually lie in a compact set. The second condition of continuity is trivially satisfied in both classification and regression as the function  $\int_{\mathcal{Y}} p_\theta(\hat{y}|y)p(y) d\mu(y)$  is continuous in  $\theta$  once  $\hat{y}$  is fixed. The third condition is trivially satisfied for classification (finite valued  $y$ ). In the case of regression due to conditions 1,2 (compactness and semi-continuity) we can replace the quantifier  $\forall \theta$  with a particular value  $\theta' \in \Theta$  representing worst case situation in the bound of the logarithm difference. Then, the bound  $K$  may be realized by the difference of log terms (with respect to that worst case  $\theta'$ ) whose expectation converges to the KL divergence which in turn is never  $\infty$  for Gaussian distributions or its derivatives. The fourth condition of measurability follows as  $p_\theta$  is specified in terms of compositions, summations, multiplications, and point-wise limits of well-known measurable functions.

The fifth condition of identifiability states that if  $p_\theta(\hat{y})$  and  $p_{\theta_0}(\hat{y})$  are identical as functions i.e., they are identical for every value of  $\hat{y}$ , then necessarily  $\theta = \theta_0$ . This condition does not hold in general and needs to be verified in each one of the special cases.

We start with establishing consistency in the case of classification where we rely on a symmetric noise model (17). The non-symmetric case (15) is more complicated and is treated afterwards. We conclude the consistency discussion with an examination of the regression case.

#### 2.4.1 Consistency of Classification Risk Estimation

**Proposition 2.** *Let  $f_1, \dots, f_k$  be classifiers  $f_i : \mathcal{X} \rightarrow \mathcal{Y}$ ,  $|\mathcal{Y}| = l$ , with conditionally independent noise processes described by (17). If the classifiers are weak learners i.e.,  $1/l < 1 - \text{err}(f_i) < 1$  and  $p(y)$  is not uniform the unsupervised collaborative diagnosis model is identifiable.*

**Corollary 2.4.1.** *Let  $f_1, \dots, f_k$  be classifiers  $f_i : \mathcal{X} \rightarrow \mathcal{Y}$  with  $|\mathcal{Y}| = l$  and noise processes described by (17). If the classifiers are weak learners i.e.,  $1/l < 1 - \text{err}(f_i) < 1$ , and  $p(y)$  is not uniform the unsupervised non-collaborative diagnosis model is identifiable.*

*Proof.* Proving identifiability in the non-collaborative case proceeds by invoking Proposition 2 (whose proof is given below) with  $k = 1$  separately for each classifier. The conditional independence assumption in Proposition 2 becomes redundant in this case of a single classifier, resulting in identifiability of  $p_{\theta_j}(\hat{y}_j)$  for each  $j = 1, \dots, k$   $\square$

**Corollary 2.4.2.** *Under the assumptions of Proposition 2 or Corollary 2.4.1 the unsupervised maximum likelihood estimator is consistent i.e.,*

$$P \left( \lim_{n \rightarrow \infty} \hat{\theta}_n^{\text{mle}}(\hat{y}^{(1)}, \dots, y^{(n)}) = (\theta_1^{\text{true}}, \dots, \theta_k^{\text{true}}) \right) = 1.$$

Consequentially, assuming that  $R(f_j) = g_j(\theta)$ ,  $j = 1, \dots, k$  with continuous  $g_j$  we also have

$$P \left( \lim_{n \rightarrow \infty} \hat{R}(f_j; y^{(1)}, \dots, y^{(n)}) = R(f_j), \quad \forall j = 1, \dots, k \right) = 1.$$

*Proof.* Proposition 2 or Corollary 2.4.1 establishes identifiability, which in conjunction with Proposition 1 proves the corollary.  $\square$

*Proof. (for Proposition 2)* We prove identifiability by induction on  $k$ . In the base case of  $k = 1$ , we have a set of  $l$  equations, corresponding to  $i = 1, 2, \dots, l$ ,

$$\begin{aligned} p_\theta(\hat{y}_1 = i) &= p(y = i)\theta_1 + \left( \sum_{j \neq i} p(y = j) \right) \frac{(1 - \theta_1)}{(l - 1)} \\ &= p(y = i)\theta_1 + (1 - p(y = i)) \frac{(1 - \theta_1)}{(l - 1)} \\ &= \frac{\theta_1(lp(y = i) - 1) + 1 - p(y = i)}{(l - 1)} \end{aligned}$$

from which we can see that if  $\eta \neq \theta$  and  $p(y = i) \neq 1/l$  then  $p_\theta(\hat{y}_1) \neq p_\eta(\hat{y}_1)$ . This proves identifiability for the base case of  $k = 1$ .

Next, we assume identifiability holds for  $k$  and prove that it holds for  $k + 1$ . We do so by deriving a contradiction from the assumption that identifiability holds for  $k$  but not for  $k + 1$ . We denote the parameters corresponding to the  $k$  labelers by the vectors  $\theta, \eta \in [0, 1]^k$  and the parameters corresponding the additional  $k + 1$  labeler by  $\theta_{k+1}, \eta_{k+1}$ .

In the case of  $k$  classifiers we have

$$p_\theta(\hat{y}_1, \dots, \hat{y}_k) = \sum_{i=1}^l p_\theta(\hat{y}_1, \dots, \hat{y}_k | y = i) p(y = i) = \sum_{i=1}^l G(\mathcal{A}_i, \theta)$$

where

$$\begin{aligned} G(\mathcal{A}_i, \theta) &\stackrel{\text{def}}{=} p(y = i) \prod_{j \in \mathcal{A}_i} \theta_j \cdot \prod_{j \notin \mathcal{A}_i} \frac{(1 - \theta_j)}{(l - 1)}. \\ \mathcal{A}_i &\stackrel{\text{def}}{=} \{j \in \{1, 2, \dots, k\} : \hat{y}_j = i\}. \end{aligned}$$

Note that the  $\mathcal{A}_1, \dots, \mathcal{A}_l$  form a partition of  $\{1, \dots, k\}$  i.e., they are disjoint and their union is  $\{1, \dots, k\}$ .

In order to have unidentifiability for the  $k + 1$  classifiers we need  $(\theta, \theta_{k+1}) \neq (\eta, \eta_{k+1})$  and the following  $l$  equations (corresponding to  $\hat{y}_{k+1} = 1, 2, \dots, l$ ) to hold for any  $\hat{y}_1, \dots, \hat{y}_k$

which corresponds to any partition  $\mathcal{A}_1, \dots, \mathcal{A}_l$

$$\begin{aligned}
\theta_{k+1}G(\mathcal{A}_1, \theta) + \frac{(1 - \theta_{k+1})}{(l - 1)} \sum_{i \neq 1} G(\mathcal{A}_i, \theta) &= \eta_{k+1}G(\mathcal{A}_1, \eta) + \frac{(1 - \eta_{k+1})}{(l - 1)} \sum_{i \neq 1} G(\mathcal{A}_i, \eta) \\
\theta_{k+1}G(\mathcal{A}_2, \theta) + \frac{(1 - \theta_{k+1})}{(l - 1)} \sum_{i \neq 2} G(\mathcal{A}_i, \theta) &= \eta_{k+1}G(\mathcal{A}_2, \eta) + \frac{(1 - \eta_{k+1})}{(l - 1)} \sum_{i \neq 2} G(\mathcal{A}_i, \eta) \\
&\vdots \\
\theta_{k+1}G(\mathcal{A}_l, \theta) + \frac{(1 - \theta_{k+1})}{(l - 1)} \sum_{i \neq l} G(\mathcal{A}_i, \theta) &= \eta_{k+1}G(\mathcal{A}_l, \eta) + \frac{(1 - \eta_{k+1})}{(l - 1)} \sum_{i \neq l} G(\mathcal{A}_i, \eta).
\end{aligned} \tag{39}$$

We consider two cases in which  $(\theta, \theta_{k+1}) \neq (\eta, \eta_{k+1})$ : (a)  $\theta \neq \eta$ , and (b)  $\theta = \eta, \theta_{k+1} \neq \eta_{k+1}$ . In the case of (a) we add the  $l$  equations above which marginalizes  $\hat{y}_{k+1}$  out of  $p_\theta(\hat{y}_1, \dots, \hat{y}_k, \hat{y}_{k+1})$  and  $p_\eta(\hat{y}_1, \dots, \hat{y}_k, \hat{y}_{k+1})$  to provide

$$\sum_{i=1}^l G(\mathcal{A}_i, \theta) = \sum_{i=1}^l G(\mathcal{A}_i, \eta) \tag{40}$$

which together with  $\theta \neq \eta$  contradicts the identifiability for the case of  $k$  classifiers.

In case (b) we have from the  $l$  equations above

$$\begin{aligned}
\theta_{k+1}G(\mathcal{A}_t, \theta) + \frac{1 - \theta_{k+1}}{l - 1} \left( \sum_{i=1}^l G(\mathcal{A}_i, \theta) - G(\mathcal{A}_t, \theta) \right) \\
= \eta_{k+1}G(\mathcal{A}_t, \eta) + \frac{1 - \eta_{k+1}}{l - 1} \left( \sum_{i=1}^l G(\mathcal{A}_i, \eta) - G(\mathcal{A}_t, \eta) \right)
\end{aligned}$$

for any  $t \in \{1, \dots, l\}$  which simplifies to

$$0 = (\theta_{k+1} - \eta_{k+1}) \left( lG(\mathcal{A}_t, \theta) - \sum_{i=1}^l G(\mathcal{A}_i, \theta) \right) \quad t = 1, \dots, k. \tag{41}$$

As we assume at this point that  $\theta_{k+1} \neq \eta_{k+1}$  the above equality entails

$$lG(\mathcal{A}_t, \theta) = \sum_{i=1}^l G(\mathcal{A}_i, \theta). \tag{42}$$

We show that (42) cannot hold by examining separately the cases  $p(y = t) > 1/l$  and  $p(y = t) < 1/l$ . Recall that there exists a  $t$  for which  $p(y = t) \neq 1/l$  since the proposition requires that  $p(y)$  is not uniform.

If  $p(y = t) > 1/l$  we choose  $\mathcal{A}_t = \{1, \dots, k\}$  and obtain

$$\begin{aligned}
lp(y = t) \prod_{j=1}^k \theta_j &= \sum_{i \neq t} p(y = i) \prod_{j=1}^k \frac{1 - \theta_j}{l - 1} + p(y = t) \prod_{j=1}^k \theta_j \\
(l - 1)p(y = t) \prod_{j=1}^k \theta_j &= (1 - p(y = t)) \prod_{j=1}^k \frac{1 - \theta_j}{l - 1} \\
p(y = t) \prod_{j=1}^k \theta_j &= \frac{(1 - p(y = t))}{(l - 1)} \prod_{j=1}^k \frac{1 - \theta_j}{l - 1}
\end{aligned}$$

which cannot hold as the term on the left hand side is necessarily larger than the term on the right hand side (if  $p(y = t) > 1/l$  and  $\theta_j > 1/l$ ). In the case  $p(y = t) < 1/l$  we choose  $\mathcal{A}_s = \{1, \dots, k\}$ ,  $s \neq t$  to obtain

$$\begin{aligned}
lp(y = t) \prod_{j=1}^k \frac{1 - \theta_j}{l - 1} &= \sum_{i \neq s} p(y = i) \prod_{j=1}^k \frac{1 - \theta_j}{l - 1} + p(y = s) \prod_{j=1}^k \theta_j \\
(lp(y = t) - p(y \neq s)) \prod_{j=1}^k \frac{1 - \theta_j}{l - 1} &= p(y = s) \prod_{j=1}^k \theta_j
\end{aligned}$$

which cannot hold as the term on the left hand side is necessarily smaller than the term on the right hand side (if  $p(y = t) < 1/l$  and  $\theta_j > 1/l$ ).

Since we derived a contradiction to the fact that we have  $k$ -identifiability but not  $k + 1$  identifiability, the induction step is proven which establishes identifiability for any  $k \geq 1$ .  $\square$

The conditions asserted above that  $p(y) \neq 1/l$  and  $1/l < 1 - \text{err}(f_i) < 1$  are intuitive. If they are violated a certain symmetry may emerge which renders the model non-identifiable and the MLE estimator not consistent.

In the case of the non-collaborative estimation for binary classification with the non-symmetric noise model, the matrix  $\theta$  in (15) is a  $2 \times 2$  matrix with two degrees of freedom as each row sums to one. In particular we have  $\theta_{11} = p_\theta(\hat{y} = 1|y = 1)$ ,  $\theta_{12} = p_\theta(\hat{y} = 1|y = 2)$ ,  $\theta_{21} = p_\theta(\hat{y} = 2|y = 1)$ ,  $\theta_{22} = p_\theta(\hat{y} = 2|y = 2)$  with the overall risk  $R(f) =$

$1 - \theta_{11}p(y = 1) - \theta_{22}p(y = 2)$ . Unfortunately, the matrix  $\theta$  is not identifiable in this case and neither is the scalar parameter  $\theta_{11}p(y = 1) + \theta_{22}p(y = 2)$  that can be used to characterize the risk.

We can, however, obtain a consistent estimator for  $\theta$  (and therefore for  $R(f)$ ) by first showing that the parameter  $\theta_{11}p(y = 1) - \theta_{22}p(y = 2)$  is identifiable and then taking the intersection of two such estimators.

**Lemma 2.4.3.** In the case of the non-collaborative estimation for binary classification with the non-symmetric noise model and  $p(y) \neq 0$ , the parameter  $\theta_{11}p(y = 1) - \theta_{22}p(y = 2)$  is identifiable.

*Proof.* For two different parameterizations  $\theta, \eta$  we have

$$p_{\theta}(\hat{y} = 1) = p(y = 1)\theta_{11} + (1 - p(y = 1))(1 - \theta_{22}) \quad (43)$$

$$p_{\theta}(\hat{y} = 2) = p(y = 1)(1 - \theta_{11}) + (1 - p(y = 1))\theta_{22} \quad (44)$$

and

$$p_{\eta}(\hat{y} = 1) = p(y = 1)\eta_{11} + (1 - p(y = 1))(1 - \eta_{22}) \quad (45)$$

$$p_{\eta}(\hat{y} = 2) = p(y = 1)(1 - \eta_{11}) + (1 - p(y = 1))\eta_{22}. \quad (46)$$

Equating the two Equations (43) and (45) we have

$$p(y = 1)(\theta_{11} + \theta_{22}) + 1 - p(y = 1) - \theta_{22} = p(y = 1)(\eta_{11} + \eta_{22}) + 1 - p(y = 1) - \eta_{22}$$

$$p(y = 1)\theta_{11} - (1 - p(y = 1))\theta_{22} = p(y = 1)\eta_{11} - (1 - p(y = 1))\eta_{22}$$

$$p(y = 1)\theta_{11} - p(y = 2)\theta_{22} = p(y = 1)\eta_{11} - p(y = 2)\eta_{22}$$

Similarly, equating Equation (44) and Equation (46) also results in  $p(y = 1)\theta_{11} - p(y = 2)\theta_{22} = p(y = 1)\eta_{11} - p(y = 2)\eta_{22}$ . As a result, we have

$$p_{\theta} \equiv p_{\eta} \quad \Rightarrow \quad p(y = 1)\theta_{11} - p(y = 2)\theta_{22} = p(y = 1)\eta_{11} - p(y = 2)\eta_{22}.$$

□

The above lemma indicates that we can use the maximum likelihood method to obtain a consistent estimator for the parameter  $\theta_{11}p(y = 1) - \theta_{22}p(y = 2)$ . Unfortunately the parameter  $\theta_{11}p(y = 1) - \theta_{22}p(y = 2)$  does not have a clear probabilistic interpretation and does not directly characterize the risk. As the following proposition shows we can obtain a consistent estimator for the risk  $R(f)$  if we have two populations of unlabeled data drawn from distributions with two distinct marginals  $p_1(y)$  and  $p_2(y)$ .

**Proposition 3.** *Consider the case of the non-collaborative estimation of binary classification risk with the non-symmetric noise model. If we have access to two unlabeled datasets drawn independently from two distributions with different marginals i.e.*

$$\begin{aligned} x^{(1)}, \dots, x^{(n)} &\stackrel{\text{iid}}{\sim} p_1(x) = \sum_y p(x|y)p_1(y) \\ x'^{(1)}, \dots, x'^{(m)} &\stackrel{\text{iid}}{\sim} p_2(x) = \sum_y p(x|y)p_2(y) \end{aligned}$$

*we can obtain a consistent estimator for the classification risk  $R(f)$ .*

*Proof.* Operating the classifier  $f$  on both sets of unlabeled data we get two sets of observed classifier outputs  $\hat{y}^{(1)}, \dots, \hat{y}^{(n)}, \hat{y}'^{(1)}, \dots, \hat{y}'^{(m)}$  where  $\hat{y}^{(i)} \stackrel{\text{iid}}{\sim} \sum_y p_\theta(\hat{y}|y)p_1(y)$  and  $\hat{y}'^{(i)} \stackrel{\text{iid}}{\sim} \sum_y p_\theta(\hat{y}|y)p_2(y)$ . In particular, note that the marginal distributions  $p_1(y)$  and  $p_2(y)$  are different but the parameter matrix  $\theta$  is the same in both cases as we operate the same classifier on samples from the same class conditional distribution  $p(x|y)$ .

Based on Lemma 2.4.3 we construct a consistent estimator for  $p_1(y = 1)\theta_{11} - p_1(y = 2)\theta_{22}$  by maximizing the likelihood of  $\hat{y}^{(1)}, \dots, \hat{y}^{(n)}$ . Similarly, we construct a consistent estimator for  $p_2(y = 1)\theta_{11} - p_2(y = 2)\theta_{22}$  by maximizing the likelihood of  $\hat{y}'^{(1)}, \dots, \hat{y}'^{(m)}$ . Note that  $p_1(y = 1)\theta_{11} - p_1(y = 2)\theta_{22}$  and  $p_2(y = 1)\theta_{11} - p_2(y = 2)\theta_{22}$  describe two lines in the 2-D space  $(\theta_{11}, \theta_{22})$ . Since the true value of  $\theta_{11}, \theta_{22}$  represent a point in that 2-D space belonging to both lines, it is necessarily the intersection of both lines (the lines cannot be parallel since their linear coefficients are distributions which are assumed to be different).

As  $n$  and  $m$  increase to infinity, the two estimators converge to the true parameter values. As a result, the intersection of the two lines described by the two estimators converges to the true values of  $(\theta_{11}, \theta_{22})$  thus allowing reconstruction of the matrix  $\theta$  and the risk  $R(f)$ .  $\square$

Clearly, the conditions for consistency in the asymmetric case are more restricted than in the symmetric case. However, situations such as in Proposition 3 are not necessarily unrealistic. In many cases it is possible to identify two unlabeled sets with different distributions. For example, if  $y$  denotes a medical condition, it may be possible to obtain two unlabeled sets from two different hospitals or two different regions with different marginal distribution corresponding to the frequency of the medical condition.

As indicated in the previous section, the risk estimation framework may be extended beyond non-collaborative estimation and collaborative conditionally independent estimation. In these extensions, the conditions for identifiability need to be determined separately, in a similar way to Corollary 2.4.1. A systematic way to do so may be obtained by noting that the identifiability equations

$$0 = p_{\theta}(\hat{y}_1, \dots, \hat{y}_k) - p_{\eta}(\hat{y}_1, \dots, \hat{y}_k) \quad \forall \hat{y}_1, \dots, \hat{y}_k$$

is a system of polynomial equations in  $(\theta, \eta)$ . As a result, demonstrating lack of identifiability becomes equivalent to obtaining a solution to a system of polynomial equations. Using Hilbert's Nullstellensatz theorem we have that a solution to a polynomial system exists if the polynomial system defines a proper ideal of the ring of polynomials [15]. As  $k$  increases the chance of identifiability failing decays dramatically as we have a system of  $l^k$  polynomials with  $2k$  variables. Such an over-determined system with substantially more equations than variables is very unlikely to have a solution.

These observations serve as both an interesting theoretical connection to algebraic geometry as well as a practical tool due to the substantial research in computational algebraic geometry. See [52] for a survey of computational algorithms and software associated with

systems of polynomial equations.

## 2.4.2 Consistency of Regression Risk Estimation

In this section, we prove the consistency of the maximum likelihood estimator  $\hat{\theta}^{\text{mle}}$  in the regression case. As in the classification case our proof centers on establishing identifiability.

**Proposition 4.** *Let  $f_1, \dots, f_k$  be regression models  $f_i(x) = a'_i x$  with  $y \sim N(\mu_y, \sigma_y^2)$ ,  $y = ax + \epsilon$ . Assuming that  $a \neq 0$  the unsupervised collaborative estimation model assuming conditionally independent noise processes (32) is identifiable.*

**Corollary 2.4.4.** *Let  $f_1, \dots, f_k$  be regression models  $f_i(x) = a'_i x$  with  $y \sim N(\mu_y, \sigma_y^2)$ ,  $y = ax + \epsilon$ . Assuming that  $a \neq 0$  the unsupervised non-collaborative estimation model (32) is identifiable.*

*Proof.* Proving identifiability in the non-collaborative case proceeds by invoking Proposition 4 (whose proof is given below) with  $k = 1$  separately for each regression model. The conditional independence assumption in Proposition 4 becomes redundant in this case of a single predictor, resulting in identifiability of  $p_{\theta_j}(\hat{y}_j)$  for each  $j = 1, \dots, k$ .  $\square$

**Corollary 2.4.5.** *Under the assumptions of Proposition 4 or Corollary 2.4.4 the unsupervised maximum likelihood estimator is consistent i.e.,*

$$P \left( \lim_{n \rightarrow \infty} \hat{\theta}_n^{\text{mle}}(\hat{y}^{(1)}, \dots, y^{(n)}) = (\theta_1^{\text{true}}, \dots, \theta_k^{\text{true}}) \right) = 1.$$

Consequentially, assuming that  $R(f_j) = g_j(\theta)$ ,  $j = 1, \dots, k$  with continuous  $g_j$  we also have

$$P \left( \lim_{n \rightarrow \infty} \hat{R}(f_j; y^{(1)}, \dots, y^{(n)}) = R(f_j), \quad \forall j = 1, \dots, k \right) = 1.$$

*Proof.* Proposition 4 or Corollary 2.4.4 establish identifiability, which in conjunction with Proposition 1 completes the proof.  $\square$

*Proof. (of Proposition 4).*

We will proceed, as in the case of classification, with induction on the number of predictors  $k$ . In the base case of  $k = 1$  we have derived  $p_{\theta_1}(\hat{y}_1)$  in Equation (24). Substituting in it  $\hat{y}_1 = 0$  we get

$$\begin{aligned} P_{\theta_1}(\hat{y}_1 = 0) &= \frac{1}{\theta_1 \sqrt{2\pi(\tau^2 + \sigma_y^2)}} \exp\left(\frac{\mu_y^2}{2\sigma_y^2} \left(\frac{\tau^2}{\sigma_y^2 + \tau^2} - 1\right)\right) \\ P_{\eta_1}(\hat{y}_1 = 0) &= \frac{1}{\eta_1 \sqrt{2\pi(\tau^2 + \sigma_y^2)}} \exp\left(\frac{\mu_y^2}{2\sigma_y^2} \left(\frac{\tau^2}{\sigma_y^2 + \tau^2} - 1\right)\right). \end{aligned} \quad (47)$$

The above expression leads to  $\theta_1 \neq \eta_1 \Rightarrow p_{\theta_1}(\hat{y}_1 = 0) \neq p_{\eta_1}(\hat{y}_1 = 0)$  which implies identifiability.

In the induction step we assume identifiability holds for  $k$  and we prove that it holds also for  $k + 1$  by deriving a contradiction to the assumption that it does not hold. We assume that identifiability fails in the case of  $k + 1$  due to differing parameter values i.e.,

$$p_{(\theta, \theta_{k+1})}(\hat{y}_1, \dots, \hat{y}_k, \hat{y}_{k+1}) = p_{(\eta, \eta_{k+1})}(\hat{y}_1, \dots, \hat{y}_k, \hat{y}_{k+1}) \forall \hat{y}_j \in \mathbb{R} \ j = 1, \dots, k + 1 \quad (48)$$

with  $(\theta, \theta_{k+1}) \neq (\eta, \eta_{k+1})$  where  $\theta, \eta \in \mathbb{R}^k$ . There are two cases which we consider separately: (a)  $\theta \neq \eta$  and (b)  $\theta = \eta$ .

In case (a) we marginalize both sides of (48) with respect to  $\hat{y}_{k+1}$  which leads to a contradiction to our assumption that identifiability holds for  $k$

$$\begin{aligned} \int_{-\infty}^{\infty} p_{(\theta, \theta_{k+1})}(\hat{y}_1, \dots, \hat{y}_k, \hat{y}_{k+1}) d\hat{y}_{k+1} &= \int_{-\infty}^{\infty} p_{(\eta, \eta_{k+1})}(\hat{y}_1, \dots, \hat{y}_k, \hat{y}_{k+1}) d\hat{y}_{k+1} \\ p_{\theta}(\hat{y}_1, \dots, \hat{y}_k) &= p_{\eta}(\hat{y}_1, \dots, \hat{y}_k). \end{aligned} \quad (49)$$

In case (b)  $\theta = \eta$  and  $\theta_{k+1} \neq \eta_{k+1}$ . Substituting  $\hat{y}_1 = \dots = \hat{y}_{k+1} = 0$  in (48) (see (33) for a derivation) we have

$$P_{(\theta, \theta_{k+1})}(\hat{y}_1 = 0, \dots, \hat{y}_{k+1} = 0) = P_{(\eta, \eta_{k+1})}(\hat{y}_1 = 0, \dots, \hat{y}_{k+1} = 0) \quad (50)$$

or

$$\begin{aligned} & \frac{\sqrt{\pi} \left[ \frac{1}{2} \left( \frac{1}{\sigma_y^2} + \frac{k+1}{\tau^2} \right) \right]^{-1/2}}{\tau^{k+1} (\sqrt{2\pi})^{k+2} \sigma_y \theta_{k+1} \prod_{j=1}^k \theta_j} \exp \left( \frac{\left( \frac{\mu_y}{\sigma_y^2} \right)^2}{2 \left( \frac{1}{\sigma_y^2} + \frac{k+1}{\tau^2} \right)} - \frac{\mu_y^2}{2\sigma_y^2} \right) \\ &= \frac{\sqrt{\pi} \left[ \frac{1}{2} \left( \frac{1}{\sigma_y^2} + \frac{k+1}{\tau^2} \right) \right]^{-1/2}}{\tau^{k+1} (\sqrt{2\pi})^{k+2} \sigma_y \eta_{k+1} \prod_{j=1}^k \eta_j} \exp \left( \frac{\left( \frac{\mu_y}{\sigma_y^2} \right)^2}{2 \left( \frac{1}{\sigma_y^2} + \frac{k+1}{\tau^2} \right)} - \frac{\mu_y^2}{2\sigma_y^2} \right) \end{aligned}$$

which cannot hold if  $\theta = \eta$  but  $\theta_{k+1} \neq \eta_{k+1}$ .  $\square$

## 2.5 Asymptotic Variance of $\hat{\theta}_n^{\text{mle}}$ and $\hat{R}$

A standard result from statistics is that the MLE has an asymptotically normal distribution with mean vector  $\theta^{\text{true}}$  and variance matrix  $(nJ(\theta^{\text{true}}))^{-1}$ , where  $J(\theta)$  is the  $r \times r$  Fisher information matrix

$$J(\theta) = \mathbf{E}_{p_\theta} \{ \nabla \log p_\theta(\hat{y}) (\nabla \log p_\theta(\hat{y}))^\top \} \quad (51)$$

with  $\nabla \log p_\theta(\hat{y})$  represents the  $r \times 1$  gradient vector of  $\log p_\theta(\hat{y})$  with respect to  $\theta$ . Stated more formally, we have the following convergence in distribution as  $n \rightarrow \infty$  [20]

$$\sqrt{n} (\hat{\theta}_n^{\text{mle}} - \theta_0) \rightsquigarrow N(0, J^{-1}(\theta^{\text{true}})). \quad (52)$$

It is instructive to consider the dependency of the Fisher information matrix, which corresponds to the asymptotic estimation accuracy, on  $n, k, p(y), \theta^{\text{true}}$ .

In the case of classification considering (17) with  $k = 1$  and  $\mathcal{Y} = \{1, 2\}$  it can be shown that

$$J(\theta) = \frac{\alpha(2\alpha - 1)^2}{(\theta(2\alpha - 1) - \alpha + 1)^2} - \frac{(2\alpha - 1)^2(\alpha - 1)}{(\alpha - \theta(2\alpha - 1))^2} \quad (53)$$

where  $\alpha = P(y = 1)$ . As Figure 4 (right) demonstrates, the asymptotic accuracy of the MLE (as indicated by  $J$ ) tends to increase with the degree of non-uniformity of  $p(y)$ . Recall that since identifiability fails for a uniform  $p(y)$  the risk estimate under a uniform  $p(y)$  is not consistent. The above derivation (53) is a quantification of that fact reflecting the added

difficulty in estimating the risk as we move closer to a uniform label distribution  $\alpha \rightarrow 1/2$ . The dependency of the asymptotic accuracy on  $\theta^{\text{true}}$  is more complex, tending to favor  $\theta^{\text{true}}$  values close to 1 or 0.5. Figure 4 (left) displays the empirical accuracy of the estimator as a function of  $p(y)$  and  $\theta^{\text{true}}$  and shows remarkable similarity to the contours of the Fisher information (see Section 2.7 for more details on the experiments). In particular, whenever the estimation error is high the asymptotic variance of the estimator is high (or equivalently, the Fisher information is low). For instance, the top contours in the left panel have smaller estimation error on the top right than in the top left. Similarly, the top contours in the right panel have smaller asymptotic variance on the top right than on the top left. We thus conclude that the Fisher information provides practical, as well as theoretical insight into the estimation accuracy.

Similar calculations of  $J(\theta^{\text{true}})$  for collaborative classification case or for the regression case result in more complicated but straightforward derivations. It is important to realize that consistency is ensured for any identifiable  $\theta^{\text{true}}, p(y)$ . The value  $(J(\theta^{\text{true}}))^{-1}$  is the constant dominating that consistency convergence.

A similar distributional analysis can be derived for the risk estimator. Applying Cramer's theorem [20] to  $\hat{R}(f_j) = g_j(\hat{\theta}^{\text{MLE}})$ ,  $j = 1, \dots, k$  and (52) we have

$$\sqrt{n}(\hat{R}(f) - R(f)) \rightsquigarrow N(0, \nabla g(\theta^{\text{true}}) J(\theta^{\text{true}}) \nabla g(\theta^{\text{true}})^\top) \quad (54)$$

where  $R(f)$ ,  $\hat{R}(f)$  are the vectors of true risk and risk estimates for the different predictors  $f_1, \dots, f_k$  and  $\nabla g(\theta^{\text{true}})$  is the Jacobian matrix of the mapping  $g = (g_1, \dots, g_k)$  evaluated at  $\theta^{\text{true}}$ .

For example, in the case of classification with  $k = 1$  we have  $R(f_j) = 1 - \theta_j$  and the Jacobian matrix is  $-1$ , leading to an identical asymptotic distribution to that of the MLE (52)-(53)

$$\sqrt{n}(\hat{R}(f) - R(f)) \rightsquigarrow N\left(0, \left(\frac{\alpha(2\alpha - 1)^2}{(\theta(2\alpha - 1) - \alpha + 1)^2} - \frac{(2\alpha - 1)^2(\alpha - 1)}{(\alpha - \theta(2\alpha - 1))^2}\right)^{-1}\right). \quad (55)$$

## 2.6 Optimization Algorithms

Recall that we obtained closed forms for the likelihood maximizers in the cases of non-collaborative estimation for binary classifiers and non-collaborative estimation for one dimensional regression models. The lack of closed form maximizers in the other cases necessitates iterative optimization techniques.

One class of technique for optimizing nonlinear loglikelihoods is the class of gradient based methods such as gradient descent, conjugate gradients, and quasi Newton methods. These techniques proceed iteratively following a search direction; they often have good performance and are easy to derive. The main difficulty with their implementation is the derivation of the loglikelihood and its derivatives. For example, in the case of collaborative estimation of classification ( $l \geq 2$ ) with symmetric noise model and missing values the loglikelihood gradient is

$$\frac{\partial \ell}{\partial \theta_j} = \sum_{i=1}^n \frac{\sum_{y^{(i)}} p(y^{(i)}) \sum_{r:\beta_{ri}=0} \sum_{\hat{y}_r^{(i)}} \prod_{p \neq j} h_{pi} (I(\hat{y}_j^{(i)} = y^{(i)}) - \theta_j) ((l-1)\theta_j)^{I(\hat{y}_j^{(i)}=y^{(i)})-1} (1-\theta_j)^{-I(\hat{y}_j^{(i)}=y^{(i)})}}{\sum_{y^{(i)}} p(y^{(i)}) \sum_{r:\beta_{ri}=0} \sum_{\hat{y}_r^{(i)}} \prod_{p=1}^k h_{pi}}$$

$$h_{pi} = \theta_p^{I(\hat{y}_p^{(i)}=y^{(i)})} \left( \frac{1-\theta_p}{l-1} \right)^{I(\hat{y}_p^{(i)} \neq y^{(i)})} \quad (56)$$

Similar derivations may be obtained in the other cases in a straightforward manner.

An alternative iterative optimization technique for finding the MLE is expectation maximization (EM). The derivation of the EM update equations is again relatively straightforward. For example in the above case of collaborative estimation of classification ( $l \geq 2$ ) with symmetric noise model and missing values the EM update equations are

$$\theta^{(t+1)} = \arg \max_{\theta} \sum_{i=1}^n \sum_{y^{(i)}} \sum_{r:\beta_{ri}=0} \sum_{\hat{y}_r^{(i)}} q^{(t)}(\hat{y}_r^{(i)}, y^{(i)}) \sum_{j=1}^k \log p_j(\hat{y}_j^{(i)} | y^{(i)}) \quad (57)$$

$$= \frac{1}{n} \sum_{i=1}^n \sum_{y^{(i)}} \sum_{r:\beta_{ri}=0} \sum_{\hat{y}_r^{(i)}} q^{(t)}(\hat{y}_r^{(i)}, y^{(i)}) I(\hat{y}_j^{(i)} = y^{(i)})$$

$$q^{(t)}(\hat{y}_r^{(i)}, y^{(i)}) = \frac{p(y^{(i)}) \prod_{j=1}^k p_j(\hat{y}_j^{(i)} | y^{(i)}, \theta^{(t)})}{\sum_{y^{(i)}} \sum_{r:\beta_{ri}=0} \sum_{\hat{y}_r^{(i)}} p(y^{(i)}) \prod_{j=1}^k p_j(\hat{y}_j^{(i)} | y^{(i)}, \theta^{(t)})}$$

where  $q^{(t)}$  is the conditional distribution defining the EM bound over the loglikelihood function.

If all the classifiers are always observed i.e.,  $\beta_{ri} = 1 \forall r, i$  Equation (78) reverts to (32), and the loglikelihood and its gradient may be efficiently computed in  $O(nlk^2)$ . In the case of missing classifier outputs a naive computation of the gradient or EM step is exponential in the number of missing values  $R = \max_i \sum_r \beta_{ri}$ . This, however, can be improved by careful dynamic programming. For example, the nested summations over the unobserved values in the gradient may be computed using a variation of the elimination algorithm in  $O(nlk^2R)$  time.

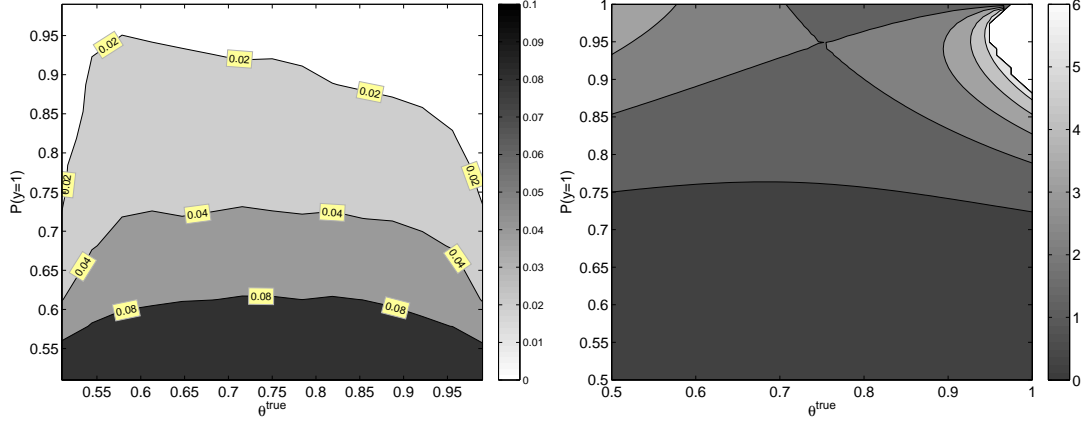
## 2.7 Empirical Evaluation

We start with some experiments demonstrating our framework using synthetic data. These experiments are meant to examine the behavior of the estimators in a controlled setting. We then describe some experiments using several real world datasets. In these experiments we examine the behavior of the estimators in an uncontrolled setting where some of the underlying assumptions may be violated. In most of the experiments we consider the mean absolute error (mae) or the  $\ell_1$  error as a metric that measures the estimation quality

$$\text{mae}(\hat{\theta}^{\text{mle}}, \theta^{\text{true}}) = \frac{1}{k} \sum_{i=1}^k |\theta_i^{\text{true}} - \hat{\theta}_i^{\text{mle}}|. \quad (58)$$

In the non-collaborative case (which is equivalent to the collaborative case with  $k = 1$ ) this translates into the absolute deviation of the estimated parameter from the true parameter.

In Figure 4 (left) we display  $\text{mae}(\hat{\theta}^{\text{mle}}, \theta^{\text{true}})$  for classification with  $k = 1$  as a function of  $\theta^{\text{true}}$  and  $p(y)$  for  $n = 500$  simulated data points. The estimation error, while overall relatively small, decays as  $p(y)$  diverges from the uniform distribution. The dependency on  $\theta^{\text{true}}$  indicates that the error is worst for  $\theta^{\text{true}}$  around 0.75 and it decays as  $|\theta^{\text{true}} - 0.75|$  increases with a larger decay attributed to higher  $\theta^{\text{true}}$ . These observations are remarkably consistent with the developed theory as Figure 4 (right) shows by demonstrating the value

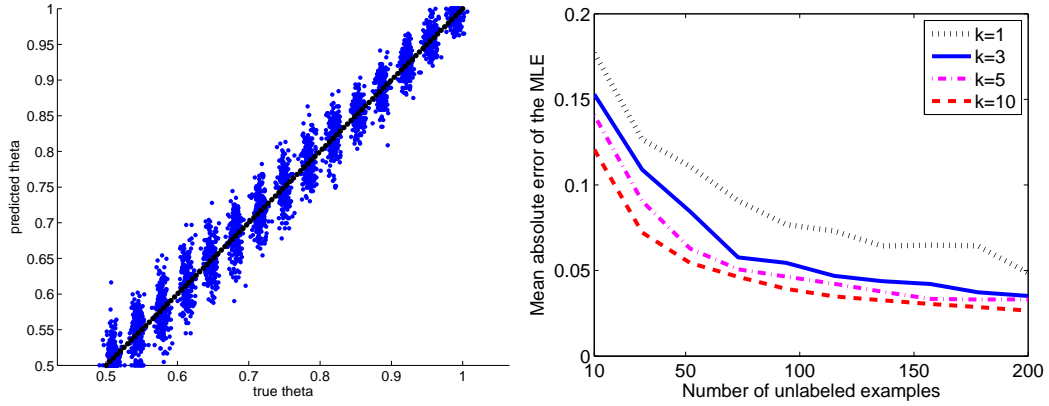


**Figure 4:** Left: Average value of  $|\hat{\theta}_n^{\text{mle}} - \theta^{\text{true}}|$  as a function of  $\theta^{\text{true}}$  and  $p(y = 1)$  for  $k = 1$  classifier and  $n = 500$  (computed over a uniform spaced grid of  $15 \times 15$  points). The plot illustrates the increased accuracy obtained by a less uniform  $P(y)$ . Right: Fisher information  $J(\theta)$  for  $k = 1$  as a function of  $\theta^{\text{true}}$  and  $P(y)$ . The asymptotic variance of the estimator is  $J^{-1}(\theta)$  which closely matches the experimental result in the left panel.

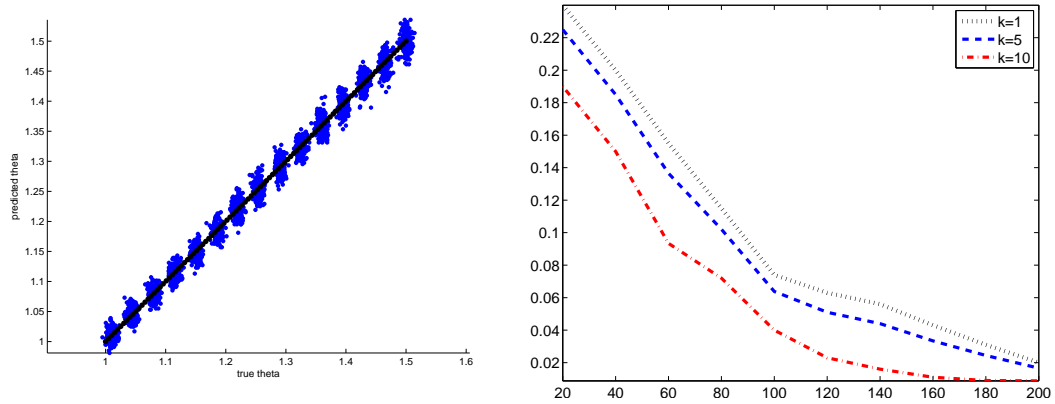
of the inverse asymptotic variance  $J(\theta)$  which agrees nicely with the empirical measurement in the left panel.

Figure 5 (left) contains a scatter plot contrasting values of  $\theta^{\text{true}}$  and  $\hat{\theta}^{\text{mle}}$  for  $k = 1$  classifier and  $p(y = 1) = 0.8$ . The estimator was constructed based on 500 simulated data points. We observe a symmetric Gaussian-like distribution of estimated values  $\hat{\theta}^{\text{mle}}$ , conditioned on specific values of  $\theta^{\text{true}}$ . This is in agreement with the theory predicting an asymptotic Gaussian distribution for the mle, centered around the true value  $\theta^{\text{true}}$ . A similar observation is made in Figure 6 (left) which contains a similar scatter plot in the regression case ( $k = 1, \sigma_y = 1, n = 1000$ ). In both figures, the striped effect is due to selection of  $\theta^{\text{true}}$  over a discrete grid with a small perturbation for increased visibility. Similar plots of larger and smaller  $n$  values (not shown) verify that the variation of  $\hat{\theta}^{\text{mle}}$  around  $\theta^{\text{true}}$  decreases as  $n$  increases. This agrees with the theory that indicates a  $O(n^{-1})$  rate of decay for the variance of the asymptotic distribution.

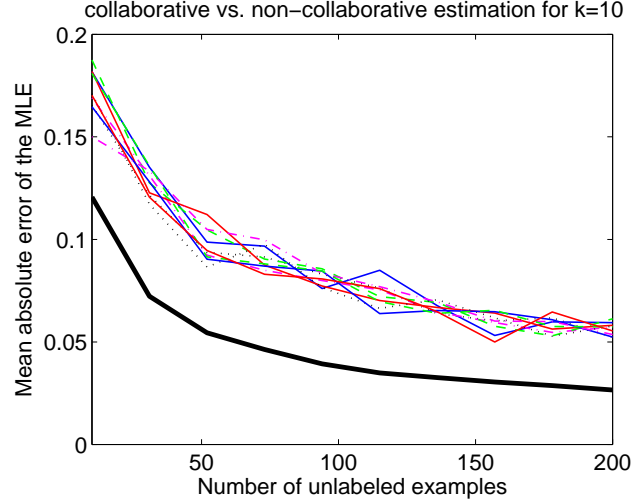
Figures 5 and 6 (right) show the  $\text{mae}(\hat{\theta}^{\text{mle}}, \theta^{\text{true}})$  for various  $k$  values in classification and regression, respectively. In classification,  $\hat{\theta}^{\text{mle}}$  was obtained by sampling data from  $p(y = 1) = 0.75 = \theta_i^{\text{true}}, \forall i$ . In regression, the data was sampled from the regression



**Figure 5:** Left: Scatter plot contrasting the true and predicted values of  $\theta$  in the case of a single classifier  $k = 1$ ,  $p(y = 1) = 0.8$ , and  $n = 500$  unlabeled examples. The displayed points were perturbed for improved visualization and the striped effect is due to empirical evaluation over a discrete grid of  $\theta^{\text{true}}$  values. Right:  $\text{mae}(\hat{\theta}^{\text{mle}}, \theta^{\text{true}})$  as a function of the number of unlabeled examples for different number of classifiers ( $\theta_i^{\text{true}} = p(y = 1) = 0.75$ ) in the collaborative case. The estimation error decreases as more classifiers are used due to the collaborative nature of the estimation process.



**Figure 6:** Left: Scatter plot contrasting the true and predicted values of  $\theta$  in the case of a single regression model  $k = 1$ ,  $\sigma_y = 1$ , and  $n = 1000$  unlabeled examples. The displayed points were perturbed for improved visualization and the striped effect is due to empirical evaluation over a discrete grid of  $\theta^{\text{true}}$  values. Right:  $\text{mae}(\hat{\theta}^{\text{mle}}, \theta^{\text{true}})$  as a function of the number of unlabeled examples for different number of regression models ( $\theta_i^{\text{true}} = \sigma_y = 1$ ) in the collaborative case. The estimation error decreases as more regression models are used due to the collaborative nature of the estimation process.

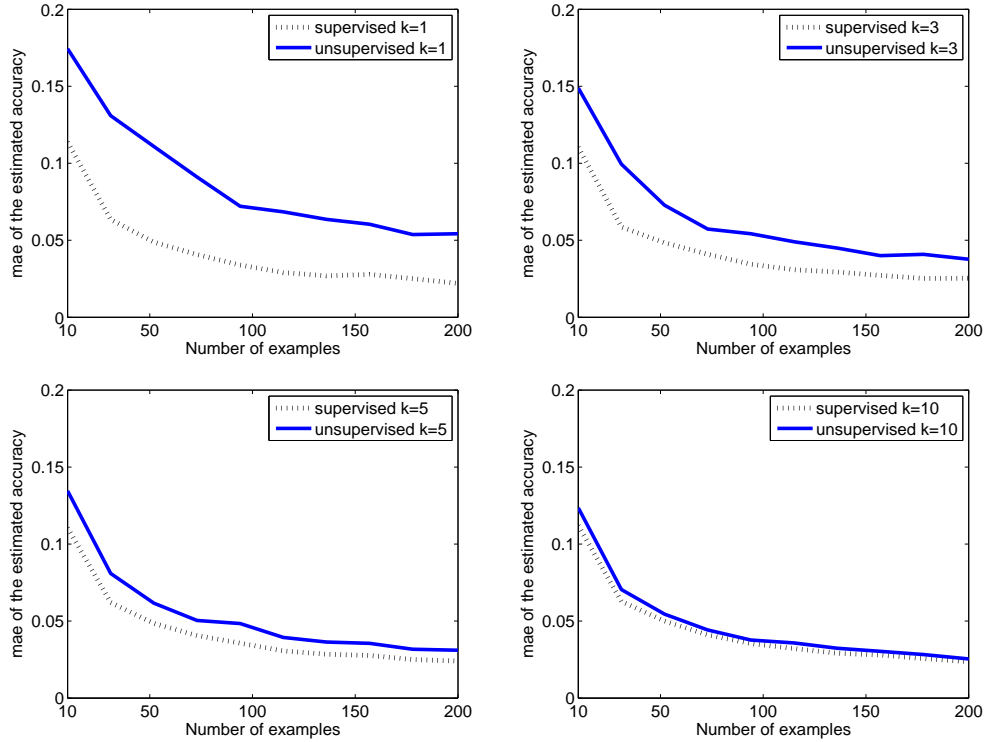


**Figure 7:** Comparison of collaborative and non-collaborative estimation for  $k = 10$  classifiers.  $\text{mae}(\hat{\theta}^{\text{mle}}, \theta^{\text{true}})$  as a function of  $n$  is reported for  $\theta_i^{\text{true}} = 0.75 \forall k_i$  and  $P(y = 1) = 0.75$ . The colored lines represent the estimation error for each individual classifier and the solid black line represents the collaborative estimation for all classifiers. The estimation converges to the truth faster in the collaborative case than in the non-collaborative case.

equation with  $\theta_i^{\text{true}} = 1$  and  $p(y) = N(0, 1)$ . In both cases, the mae error decays with  $n$  as expected from the consistency proof and with  $k$  as a result of the collaborative estimation effect.

To further illustrate the effect of the collaboration on the estimation accuracy, we estimated the error rates individually (non-collaboratively) for 10 predictors and compared their mae to that of the collaborative estimation case in Figure 7. This shows that each of the classifiers have a similar mae curve when non-collaborative estimation is used. However, all of these curves are higher than the collaborative mae curve (solid black line in Figure 7) demonstrating the improvement of the collaborative process.

We compare in Figure 8 the proposed unsupervised estimation framework with supervised estimation that takes advantage of labeled information to determine the classifier accuracy. We conducted this study using equal number of examples for both supervised and unsupervised cases. Clearly, this is an unfair comparison if we assume that labeled data is unavailable or is difficult to obtain. The unsupervised estimation does not perform

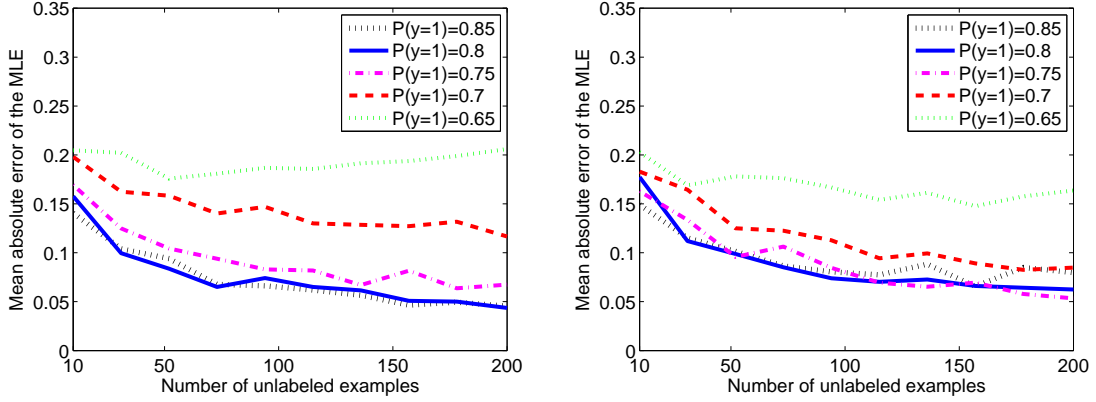


**Figure 8:** Comparison of supervised and unsupervised estimation for different values of classifiers with  $k = 1, 3, 5, 10$ . Supervised estimation uses the true labels to determine the accuracy of the classifiers whereas in the unsupervised case the estimation proceeds according to the collaborative estimation framework. Despite the fact that the supervised case uses labels the unsupervised framework reaches similar levels by increasing the number of classifiers.

as well as the supervised version especially in general. Nevertheless, the unsupervised estimation accuracy improves significantly with increasing number of classifiers and finally reaches the performance level of the supervised case due to collaborative estimation.

In Figure 9 we report the effect of misspecification of the marginal  $p(y)$  on the estimation accuracy. More specifically, we generated synthetic data using a true marginal distribution but estimated the classifier accuracy on this data assuming a misspecified marginal. Generally, the estimation framework is robust to small perturbations while over-specifying tends to hurt less than under-specifying (misspecification closer to uniform distribution).

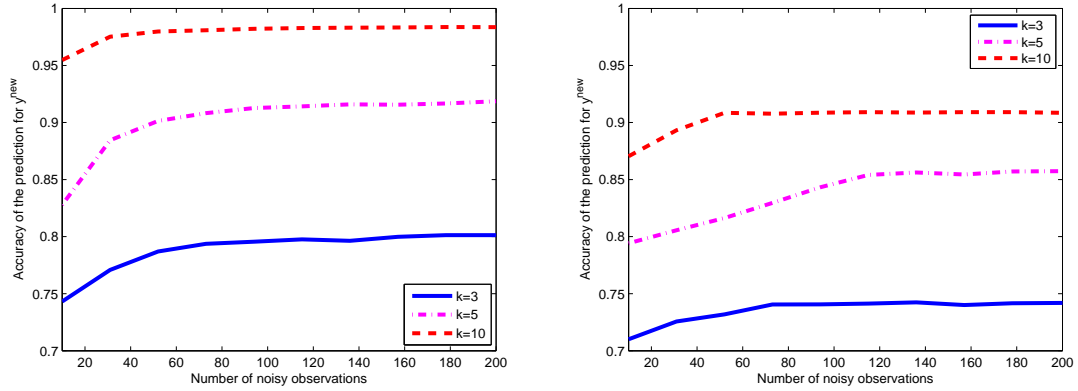
Figure 10 shows the mean prediction accuracy for the unsupervised predictor combination scheme in (12) for synthetic data. The left panel displays classification accuracy and



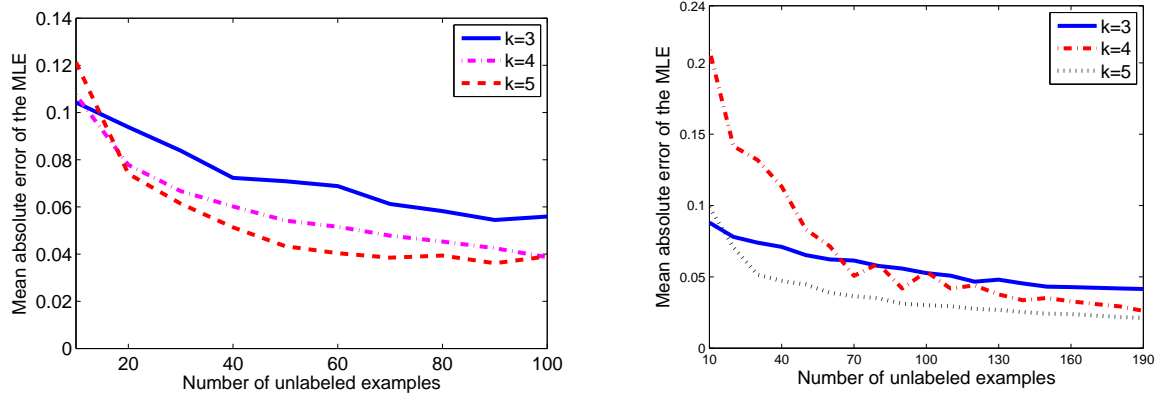
**Figure 9:** The figure compares the estimator accuracy assuming that the marginal  $p(y)$  is misspecified. The plots draw  $\text{mae}(\hat{\theta}^{\text{mle}}, \theta^{\text{true}})$  as a function of  $n$  for  $k = 1$  and  $\theta^{\text{true}} = 0.75$  when  $P^{\text{true}}(y = 1) = 0.8$  (left) and  $P^{\text{true}}(y = 1) = 0.75$  (right). Small perturbations in  $P^{\text{true}}(y)$  do not affect the results significantly; interestingly over-specifying  $P^{\text{true}}(y = 1)$  leads to more accurate estimates than under-specifying (misspecification closer to uniform distribution)

the right panel displays the regression accuracy as measured by  $1 - \frac{1}{m} \sum_{i=1}^m (\hat{y}_i^{\text{new}} - y_i^{\text{new}})^2$ . The graphs show that in both cases the accuracy increases with  $k$  and  $n$  in accordance with the theory and the risk estimation experiments. The parameter  $\theta_i^{\text{true}}$  was chosen uniformly in the range  $(0.5, 1)$ , and  $P(y = 1) = 0.75$  for classification and  $\theta_i^{\text{true}} = 0.3$ ,  $p(y) = N(0, 1)$  in the case of regression.

We also experimented with the natural language understanding dataset introduced in [51]. This data was created using the Amazon Mechanical Turk (AMT) for data annotation. AMT is an online tool that uses paid employees to complete small labeling and annotation tasks. We selected two binary tasks from this data: the textual entailment recognition (RTE) and temporal event recognition (TEMP) tasks. In the former task, the annotator is presented with two sentences for each question. He needs to decide whether the second sentence can be inferred from the first. The original dataset contains 800 sentence pairs with a total of 165 annotators. The latter task involves recognizing the temporal relation in verb-event pairs. The annotator is forced to decide whether the event described by the first verb occurs before or after the second. The original dataset contains 462 pairs and 76 annotators. In both datasets, most of the annotators have completed only a handful of tasks. Therefore, we



**Figure 10:** Mean prediction accuracy for the unsupervised predictor combination scheme in (12) for synthetic data. The left panel displays classification accuracy and the right panel displays the regression accuracy as measured by  $1 - \frac{1}{m} \sum_{i=1}^m (\hat{y}_i^{\text{new}} - y_i^{\text{new}})^2$ . The graphs show that in both cases the accuracy increases with  $k$  and  $n$  in accordance with the theory and the risk estimation experiments.



**Figure 11:**  $\text{mae}(\hat{\theta}^{\text{mle}}, \theta^{\text{true}})$  as a function of  $n$  for different number of annotators  $k$  on RTE (left) and TEMP (right) datasets. Left:  $n = 100$ ,  $P(y = 1) = 0.5$  and  $\theta^{\text{true}} = \{0.85, 0.92, 0.58, 0.5, 0.51\}$ . Right:  $n = 190$ ,  $P(y = 1) = 0.56$  and  $\theta^{\text{true}} = \{0.93, 0.92, 0.54, 0.44, 0.92\}$ . The classifiers were added in the order specified.

selected a subset of these annotators for each task such that each annotator has completed at least 100 problems and has differing accuracies. The datasets contain ground truth labels which are used solely to calculate the annotator accuracy and not used at all during the estimation process. For efficiency, we selected only the instances for which all annotators provide an answer. This resulted in  $n = 100, 190$  for RTE and TEMP, respectively.

In Figure 11 we display  $\text{mae}(\theta^{\text{true}}, \hat{\theta}^{\text{mle}})$  for these datasets as function of  $n$  for different values of  $k$ . These plots generated from real-world data show similar trend to the synthetic experiments. The estimation errors decay to 0 as  $n$  increases and generally tend to decrease as  $k$  increases. This correspondence is remarkable since two of the labelers have worse than random accuracy and since it is not clear whether the conditional independence assumption actually holds in reality for these datasets. Nevertheless, the collaborative estimation error behaves in accordance with the synthetic data experiments and the theory. This shows that the estimation framework is robust to the breakdown of the assumption that the classifier accuracy must be higher than random choice. Also, whether the conditional independence assumption holds or not is not crucial in this case.

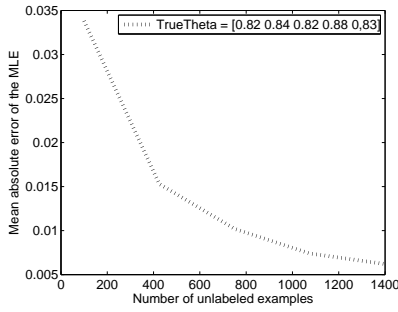
We further experimented with classifiers trained on different representations of the same dataset and estimated their error rates. We adopted the Ringnorm dataset generated by [10]. Ringnorm is a 2-class artificial dataset with 20 dimensions where each class is drawn from a multivariate normal distribution. One class has zero mean and a covariance  $\Sigma = 4I$  where  $I$  is the identity matrix. The other class has unit covariance and a mean  $\mu = (\frac{2}{\sqrt{20}}, \frac{2}{\sqrt{20}}, \dots, \frac{2}{\sqrt{20}})$ . The total size is 7400. We created 5 different representations of the data by projecting it onto mutually exclusive sets of principal components obtained by Principal Component Analysis (PCA). We trained an SVM classifier (with 2-degree polynomial kernel) [56, 26] on samples from each representation while holding out 1400 examples as the test set resulting in a total of 5 classifiers. We tested each of the 5 classifiers on the test set and used their outputs to estimate the corresponding parameters. The true labels of the test set examples were used as ground truth to calculate the mae of the mle

estimators.

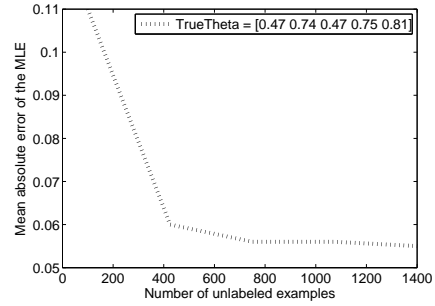
The mae curves for this dataset appear in Figure 12 as a function of the number  $n$  of unlabeled examples. When all classifiers are highly accurate (upper left panel), the collaborative unsupervised estimator is reliable, see Figure 12(a). With a mixture of weak and strong classifiers (upper right panel), the collaborative unsupervised estimator is also reliable. This is despite the fact that some of the weak classifiers in Figure 12(b) have worse than random accuracy which violates the assumptions in the consistency proposition. This shows again that the estimation framework is robust to occasional deviations from the requirement concerning better than random classification accuracies. On the other hand, as most of the classifiers become worse (bottom row), the accuracy of the unsupervised estimator decreases, in accordance with the theory developed in Sections 2.5 (recall the Fisher information contour plot).

Our experiments thus far assumed the symmetric noise model (17). Despite it not being always applicable for real world data and classifiers, it did result in good estimation accuracy in some of the cases described thus far. However, in some cases this assumption is grossly violated and the more general noise model is needed (15). For this reason, we conducted two experiments using real world data assuming the more general (15).

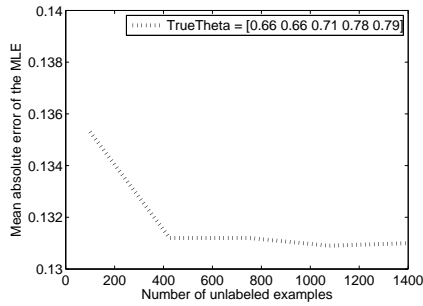
The first experiment concerned domain adaptation [9] for Amazon’s product reviews in four different product domains: books, DVDs, electronics and kitchen appliances. Each domain consists of positive ( $y = 1$ ) and negative ( $y = 2$ ) reviews with  $p(y = 1) = 0.75$ . The task was to estimate the error rates of classifiers (linear SVM [56, 26]) that are trained on 300 examples from one domain but tested on other domains. The mae values for the classification risks are displayed in Figure 15 with the columns indicating the test domain. In this case, the unsupervised non-collaborative estimator outperforms the collaborative estimator due to violation of the conditional independence assumption. Both unsupervised estimators perform substantially better than the baseline estimator that uses the training error on one domain to predict testing error on another domain.



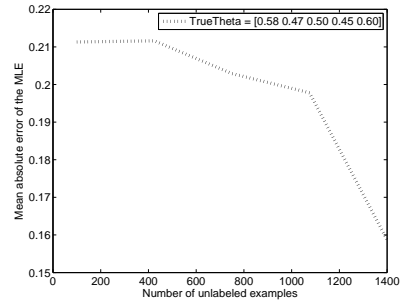
(a) Strong classifiers



(b) A mixture of strong and weak classifiers



(c) Mostly weak classifiers



(d) Very weak classifiers

**Figure 12:**  $\text{mae}(\theta^{\text{true}}, \hat{\theta}^{\text{mle}})$  as a function of the test set size on the Ringnorm dataset.  $p(y = 1) = 0.47$ , and  $\theta^{\text{true}}$  is indicated in the legend in each plot. The four panels represent mostly strong classifiers (upper left), a mixture of strong and weak classifiers (upper right), mostly weak classifiers (bottom left), and mostly very weak classifiers (bottom right). The figure shows that the framework is robust to occasional deviations from the assumption regarding better than random guess classification accuracy (upper right panel). However, as most of the classifiers become weak or very weak, the collaborative unsupervised estimation framework results in worse estimation error.

	book	dvd	kitchen	electronics	20newsgroup
training error	0.22	0.23	0.26	0.30	0.028
non-collaborative	<b>0.04</b>	<b>0.04</b>	<b>0.08</b>	<b>0.06</b>	<b>0.006</b>
collaborative	0.10	0.10	0.09	0.08	n/a

**Figure 13:**  $\text{mae}(\hat{\theta}^{\text{mle}}, \theta^{\text{true}})$  for the domain adaptation ( $n = 1000$ ,  $p(y = 1) = 0.75$ ) and 20 newsgroup ( $n = 15,000$ ,  $p(y = 1) = 0.05$  for each one-vs-all data). The unsupervised non-collaborative estimator outperforms the collaborative estimator due to violation of the conditional independence assumption. Both unsupervised estimators perform substantially better than the baseline training error rate estimator. In both cases the results were averaged over 50 random train test splits.

In the second experiment using (15) we estimated the risk (non-collaboratively) of 20 one vs. all classifiers (trained to predict one class) on the 20 newsgroup data [32]. The train set size was 1000 and the unlabeled data size was 15000. In this case the unsupervised non-collaborative estimator returned extremely accurate risk estimators. As a comparison, the risk estimates obtained from the training error are four times larger than the unsupervised MLE estimator (See Figure 15).

## CHAPTER III

### TRAINING MARGIN BASED CLASSIFIERS WITHOUT LABELS

#### 3.1 Definitions

For standard linear classifiers  $\hat{Y} = \text{sign} \sum \theta_j X_j$  with  $Y \in \{-1, +1\}$ ,  $X \in \mathbb{R}^d$  the margin is defined as

$$Y f_\theta(X) \quad \text{where} \quad f_\theta(X) \stackrel{\text{def}}{=} \sum_{j=1}^d \theta_j X_j. \quad (59)$$

Training such classifiers involves choosing a particular value of  $\theta$ . This is done by minimizing the risk or expected loss

$$R(\theta) = \mathbf{E}_{p(X,Y)} L(Y, f_\theta(X)). \quad (60)$$

Three popular examples of the loss  $L$  are

$$L_1(Y, f_\theta(X)) = \exp(-Y f_\theta(X)) \quad (61)$$

$$L_2(Y, f_\theta(X)) = \log(1 + \exp(-Y f_\theta(X))) \quad (62)$$

$$L_3(Y, f_\theta(X)) = (1 - Y f_\theta(X))_+. \quad (63)$$

that correspond to exponential loss (boosting), logloss (logistic regression) and hinge loss (SVM) respectively.

Since the risk  $R(\theta)$  depends on the unknown distribution  $p$ , it is usually replaced during training with its empirical counterpart based on a labeled training set

$$(X^{(1)}, Y^{(1)}), \dots, (X^{(n)}, Y^{(n)}) \stackrel{\text{iid}}{\sim} p \quad (64)$$

leading to the following estimator

$$\hat{\theta}_n = \arg \min_{\theta} R_n(\theta) \quad \text{where}$$

$$R_n(\theta) = \mathbf{E}_{\hat{p}(X,Y)} L(Y, f_{\theta}(X)) \quad (65)$$

$$= \frac{1}{n} \sum_{i=1}^n L(Y^{(i)}, f_{\theta}(X^{(i)})). \quad (66)$$

Note, however, that evaluating and minimizing  $R_n$  requires labeled data (64). While suitable in some cases, there are certainly situations in which labeled data is difficult or impossible to obtain.

We construct an estimator for  $R(\theta)$  using only unlabeled data, that is using

$$X^{(1)}, \dots, X^{(n)} \stackrel{\text{iid}}{\sim} p \quad (67)$$

instead of (64). Our estimator is based on the following observations. When the data is high dimensional ( $d \rightarrow \infty$ ) the quantities

$$f_{\theta}(X)|Y = y, \quad y \in \{-1, +1\} \quad (68)$$

are often normally distributed ( $f_{\theta}(X) = \langle \theta, X \rangle$  as in (59)). This phenomenon is supported by empirical evidence and may also be derived using non-iid central limit theorems. We then observe that the limit distributions of (68) may be estimated from unlabeled data (67) and that these distributions may be used to measure margin-based losses such as (61)-(63).

We examine two novel unsupervised applications: (i) estimating margin-based losses in transfer learning and (ii) training margin-based classifiers. We investigate these applications theoretically and also provide empirical results on synthetic and real-world data. Our empirical evaluation shows the effectiveness of the proposed framework in risk estimation and classifier training without any labeled data.

The consequences of estimating  $R(\theta)$  without labels are indeed profound. Label scarcity is a well known problem which has led to the emergence of semisupervised learning:

learning using a few labeled examples and many unlabeled ones. The techniques we develop lead to a new paradigm that goes beyond semisupervised learning in requiring no labels whatsoever.

### 3.2 *Unsupervised Risk Estimation*

In this section we describe in detail the proposed estimation framework and discuss its theoretical properties. Specifically, we construct an estimator for  $R(\theta)$  (60) using the unlabeled data (67).

Our estimation is based on two assumptions. The first assumption is that the label marginals  $p(Y)$  are known and that  $p(Y = 1) \neq p(Y = -1)$ . While this assumption may seem restrictive at first, there are many cases where it holds. Examples include medical diagnosis ( $p(Y)$  is the well known marginal disease frequency), handwriting recognition or OCR ( $p(Y)$  is the easily computable marginal frequencies of different letters in the English language), life expectancy prediction ( $p(Y)$  is based on marginal life expectancy tables). In these and other examples  $p(Y)$  is known with great accuracy even if labeled data is unavailable.

The second assumption is that the quantity  $f_\theta(X)|Y$  follows a normal distribution. As  $f_\theta(X)$  is a linear combination of random variables, it is frequently normal when  $X$  is high dimensional. This assumption holds empirically for many high dimensional data (see Section 3.2.1). From a theoretical perspective this assumption is motivated by the central limit theorem (CLT). The classical CLT states that  $f_\theta(X) = \sum_{i=1}^d \theta_i X_i$  is approximately normal for large  $d$  if the data components  $X_1, \dots, X_d$  are iid. A more general CLT state that  $f_\theta(X)|Y$  is asymptotically normal if  $X_1, \dots, X_d$  are independent (not necessary identically distributed). Even more general CLTs state that  $f_\theta(X)|Y$  is asymptotically normal if  $X_1, \dots, X_d$  are not independent but their dependency is limited in some way. We examine this issue in Section 3.2.1.

To derive the estimator we rewrite (60) by taking expectation with respect to  $Y$  and

$$\alpha = f_\theta(X)$$

$$\begin{aligned}
R(\theta) &= \mathbf{E}_{p(f_\theta(X), Y)} L(Y, f_\theta(X)) \\
&= \sum_{y \in \{-1, +1\}} p(y) \int_{\mathbb{R}} p(f_\theta(X) = \alpha | y) L(y, \alpha) d\alpha \\
&= p(y = 1) \int_{\mathbb{R}} p(f_\theta(X) = \alpha | y = 1) L(1, \alpha) d\alpha \\
&\quad + p(y = -1) \int_{\mathbb{R}} p(f_\theta(X) = \alpha | y = -1) L(-1, \alpha) d\alpha.
\end{aligned} \tag{69}$$

Equation (69) involves three terms  $L(y, \alpha)$ ,  $p(y)$  and  $p(f_\theta(X) = \alpha | y)$ . The loss function  $L$  is known and poses no difficulty. The second term  $p(y)$  is assumed to be known (see discussion above). The third term is normal (assuming a CLT holds cf. Section 3.2.1)

$$f_\theta(X) | y = \sum_i \theta_i X_i | y \sim N(\mu_y, \sigma_y)$$

with parameters  $\mu_y, \sigma_y, y \in \{-1, 1\}$  that are generally unknown. Note that although we do not denote it explicitly,  $\mu_y$  and  $\sigma_y$  are functions of  $\theta$ .

We conclude with estimating  $\mu = (\mu_1, \mu_{-1})$  and  $\sigma = (\sigma_1, \sigma_{-1})$  by maximizing the likelihood of (67)

$$\begin{aligned}
(\hat{\mu}^{(n)}, \hat{\sigma}^{(n)}) &= \arg \max_{\mu, \sigma} \ell_n(\mu, \sigma) \quad \text{where} \\
\ell_n(\mu, \sigma) &= \sum_{i=1}^n \log p(f_\theta(X^{(i)})) \\
&= \sum_{i=1}^n \log \sum_{y^{(i)}} p(f_\theta(X^{(i)}), y^{(i)}) \\
&= \sum_{i=1}^n \log \sum_{y^{(i)}} p(y^{(i)}) p_{\mu_{y^{(i)}}, \sigma_{y^{(i)}}}(f_\theta(X^{(i)}) | y^{(i)}).
\end{aligned} \tag{70}$$

Note that the loglikelihood (70) does not use labeled data (the label  $y^{(i)}$  is marginalized over as it is unknown). Also, the loglikelihood (70) parameter is  $\mu = (\mu_1, \mu_{-1})$  and  $\sigma = (\sigma_1, \sigma_{-1})$ , rather than the parameter  $\theta$  associated with the classifier. We consider the latter one as a fixed constant at this point.

The estimation problem (70) is equivalent to the problem of estimating the means and variances of a Gaussian mixture model where the label marginals are assumed to be known. As we show in Section 3.2.2 the estimator (70) is consistent, that is  $\lim_n(\hat{\mu}^{(n)}, \hat{\sigma}^{(n)}) = (\mu, \sigma)$  as  $n \rightarrow \infty$ , leading to the convergence of the plug-in estimate (see Section 3.2.2 for a proof)

$$P\left(\lim_{n \rightarrow \infty} \hat{R}_n(\theta) = R(\theta)\right) = 1 \quad \text{where} \quad (71)$$

$$\begin{aligned} \hat{R}_n(\theta) &= \sum_{y \in \{-1, +1\}} p(y) \\ &\times \int_{\mathbb{R}} p_{\hat{\mu}_y^{(n)}, \hat{\sigma}_y^{(n)}}(f_\theta(X) = \alpha|y) L(y, \alpha) d\alpha. \end{aligned} \quad (72)$$

### 3.2.1 Asymptotic Normality of $f_\theta(X)|Y$

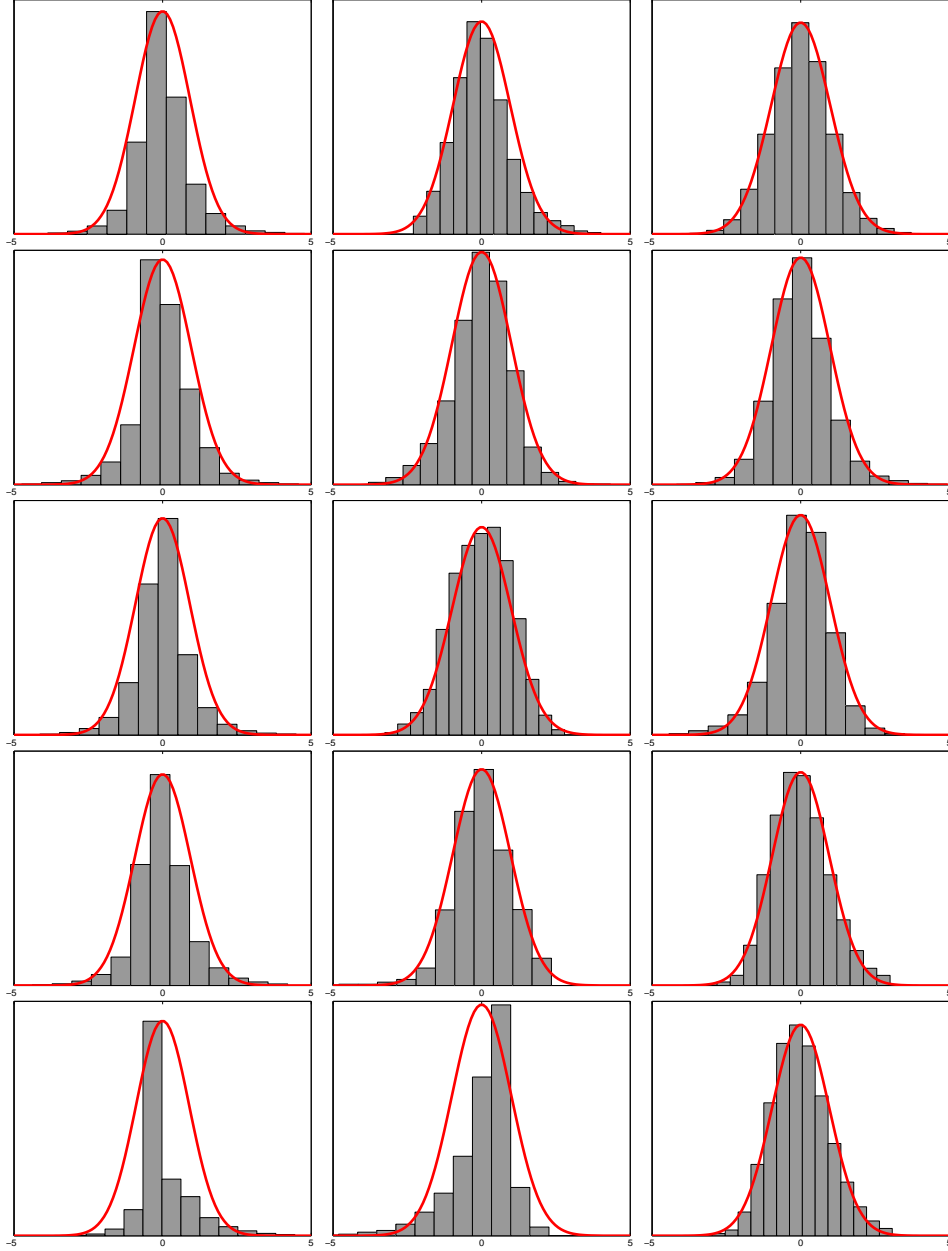
The quantity  $f_\theta(X)|Y$  is essentially a sum of  $d$  random variables which for large  $d$  is likely to be normally distributed. As we show in Figure 14 this holds in practice for text, digit images, and face images data. From a theoretical standpoint normality may be argued using a central limit theorem. We examine below three progressively more general central limit theorems and discuss whether these theorems are likely to hold in practice for high dimensional data.

The original central limit theorem states that  $\sum_{i=1}^d Z_i$  is approximately normal for large  $d$  if  $Z_i$  are iid.

**Proposition 5** (de-Moivre). *If  $Z_i, i \in \mathbb{N}$  are iid with expectation  $\mu$  and variance  $\sigma^2$  and  $\bar{Z}_d = d^{-1} \sum_{i=1}^d Z_i$  then we have the following convergence in distribution*

$$\sqrt{d}(\bar{Z}_d - \mu)/\sigma \rightsquigarrow N(0, 1) \quad \text{as } d \rightarrow \infty.$$

As a result, the quantity  $\sum_{i=1}^d Z_i$  (which is a linear transformation of  $\sqrt{d}(\bar{Z}_d - \mu)/\sigma$ ) is approximately normal for large  $d$ . This relatively restricted theorem is unlikely to hold in most practical cases as  $X_1, \dots, X_d$  are often not iid. Moreover, even if  $X_1, \dots, X_d$  are iid, the summands  $Z_i = \theta_i X_i$  are not iid.



**Figure 14:** Centered histograms of  $f_{\theta}(X)|Y = 1$  overlaid with the pdf of a fitted Gaussian for multiple  $\theta$  vectors (five rows: random  $\theta_i \sim U(-1/2, 1/2)$ , Fisher’s LDA, logistic regression,  $l_2$  regularized logistic regression, and  $l_1$  regularized logistic regression—all regularization parameters were selected by cross validation) and datasets (columns: RCV1 text data [36], MNIST digit images, and face images [42]). The fifteen panels show that even in moderate dimensionality (RCV1: 1000 top words, MNIST digits: 784 pixels, face images: 400 pixels) the assumption that  $f_{\theta}(X)|Y$  is normal holds well (except perhaps for  $l_1$  regularization in the last row which promotes sparse  $\theta$ ).

A more general CLT by Lindberg does not require that the summands  $Z_i$  be identically distributed.

**Proposition 6** (Lindberg). *For  $Z_i, i \in \mathbb{N}$  independent with expectation  $\mu_i$  and variance  $\sigma_i^2$ , and denoting  $s_d^2 = \sum_{i=1}^d \sigma_i^2$ , we have the following convergence in distribution as  $d \rightarrow \infty$*

$$s_d^{-1} \sum_{i=1}^d (Z_i - \mu_i) \rightsquigarrow N(0, 1)$$

if the following condition holds for every  $\epsilon > 0$

$$\lim_{d \rightarrow \infty} s_d^{-2} \sum_{i=1}^d \mathbf{E} (Z_i - \mu_i)^2 \mathbf{1}_{\{|Z_i - \mu_i| > \epsilon s_d\}} = 0. \quad (73)$$

This CLT is more general as it only requires that the data dimensions be independent. The condition (73) is relatively mild and specifies that contributions of each of the  $Z_i$  to the variance  $s_d$  should not dominate it. Nevertheless, the Lindberg CLT is still not satisfactory as in many cases the data dimensions are dependent.

More general CLTs replace the condition that  $Z_i, i \in \mathbb{N}$  be independent with the notion of  $m(k)$ -dependence.

**Definition 1.** The random variables  $Z_i, i \in \mathbb{N}$  are said to be  $m(k)$ -dependent if whenever  $s - r > m(k)$  the two sets  $\{Z_1, \dots, Z_r\}, \{Z_s, \dots, Z_k\}$  are independent.

An early CLT for  $m(k)$ -dependent RVs is [25]. Below is a slightly weakened version of the CLT in [6].

**Proposition 7** (Berk). *For each  $k \in \mathbb{N}$  let  $d(k)$  and  $m(k)$  be increasing sequences and suppose that  $Z_1^{(k)}, \dots, Z_{d(k)}^{(k)}$  is an  $m(k)$ -dependent sequence of random variables. If*

1.  $\mathbf{E}|Z_i^{(k)}|^2 \leq M$  for all  $i$  and  $k$
2.  $\mathbf{Var}(Z_{i+1}^{(k)} + \dots + Z_j^{(k)}) \leq (j - i)K$  for all  $i, j, k$
3.  $\lim_{k \rightarrow \infty} \mathbf{Var}(Z_1^{(k)} + \dots + Z_{d(k)}^{(k)})/d(k)$  exists and is non-zero

$$4. \lim_{k \rightarrow \infty} m^2(k)/d(k) = 0$$

then  $\frac{\sum_{i=1}^{d(k)} Z_i^{(k)}}{\sqrt{d(k)}}$  is asymptotically normal as  $k \rightarrow \infty$ .

Proposition 7 states that under mild conditions the sum of  $m(k)$ -dependent RVs is asymptotically normal. If  $m(k)$  is a constant i.e.,  $m(k) = m$ ,  $m(k)$ -dependence implies that a  $Z_i$  may only depend on its neighboring dimensions. Or in other words, dimensions that are removed from each other are independent. The full power of Proposition 7 is invoked when  $m(k)$  grows with  $k$  relaxing the independence restriction as the dimensionality grows. Intuitively, the dependency of the summands is not fixed to a certain order, but it cannot grow too rapidly.

At the end of the day, the question of whether  $f_\theta(X)|Y$  is approximately normal should be determined in practice, within a specific context. In many cases the dimensionality  $d$  is high. For example, in the case of text documents ( $X_i$  is the relative number of times word  $i$  appeared in the document)  $d$  corresponds to the vocabulary size which is typically a large number in the range  $10^3 - 10^5$ . Similarly, in the case of image classification ( $X_i$  denotes the brightness of the  $i$ -pixel) the dimensionality is on the order of  $10^2 - 10^4$ .

The question of whether such data is  $m(k)$ -dependent and whether  $f_\theta(X)|Y$  is normal is an empirical one. Figure 14 answers this question in the affirmative for three separate datasets containing text and image data. Specifically, the variable  $f_\theta(X)|Y$  is approximately normal for RCV1 data (text) [36], handwritten digit images, and face images [42]. This holds broadly both for randomly generated  $\theta$  and for  $\theta$  estimated using Fisher's LDA and logistic regression (top 3 rows). We further observe that normality holds for  $\theta$  obtained using regularized logistic regression with a broad range of regularization parameters governing the amount of sparsity. The distribution of  $f_\theta(X)|Y$  deviates from normal for radically sparse  $\theta$ , as evidenced by the histograms of  $l_1$  regularized logistic regression (last row).

Encouraged by this empirical observation and by the theoretical motivations we proceed in the next section to prove identifiability and unsupervised consistency of the risk

estimator, assuming normality of  $f_\theta(X)|Y$ .

### 3.2.2 Unsupervised Consistency

Under the assumptions specified above, in particular that  $p(y)$  is known and that  $f_\theta(X)|Y$  is normal, the plug-in estimator (72) is consistent in the unsupervised sense. In other words, the risk estimator  $\hat{R}_n$  converges to the true risk as the amount of unlabeled data increases.

We start with proving identifiability of the maximum likelihood estimator (MLE) for a mixture of two Gaussians with known mixture proportions. Invoking classical consistency results in conjunction with identifiability we show consistency of the MLE estimator for  $(\mu, \sigma)$  parameterizing the distribution of  $f_\theta(X)|Y$ . Consistency of the estimator  $\hat{R}_n$  follows.

**Definition 2.** A parametric family  $\{p_\alpha : \alpha \in A\}$  is identifiable when  $p_\alpha(x) = p_{\alpha'}(x), \forall x$  implies  $\alpha = \alpha'$ .

**Proposition 8.** *Assuming known label marginals with  $p(y = 1) \neq p(y = -1)$ , the Gaussian mixture family*

$$p_{\mu, \sigma}(x) = p(y = 1)N(x; \mu_1, \sigma_1^2) + p(y = -1)N(x; \mu_{-1}, \sigma_{-1}^2)$$

*is identifiable.*

*Proof.* It can be shown that the family of Gaussian mixture model with unknown label marginals (that is  $p(y)$  is also a parameter) is identifiable up to a permutation of the labels  $y$  [54].

We proceed by assuming with no loss of generality that  $p(y = 1) > p(y = -1)$ . The alternative case  $p(y = 1) < p(y = -1)$  may be handled in the same manner. Using the result of [54] we have that if  $p_{\mu, \sigma}(x) = p_{\mu', \sigma'}(x)$  for all  $x$ , then  $(p(y), \mu, \sigma) = (p(y), \mu', \sigma')$  up to a permutation of the labels. Since permuting the labels violates our assumption  $p(y = 1) > p(y = -1)$  we establish  $(\mu, \sigma) = (\mu', \sigma')$  proving identifiability.  $\square$

The assumption that  $p(y)$  is known is not entirely crucial. It may be relaxed by assuming that it is known whether  $p(Y = 1) > p(Y = -1)$  or  $p(Y = 1) < p(Y = -1)$ . Proving Proposition 8 under this much weaker assumption follows identical lines.

**Proposition 9.** *Under the assumptions of Proposition 8 maximizing (70) as a function of  $(\mu, \sigma) = (\mu_1, \mu_{-1}, \sigma_1, \sigma_{-1})$  provides a consistent estimator for the distributions of  $f_\theta(X)|Y = 1$  and  $f_\theta(X)|Y = -1$ . In other words, the sequence of MLE estimators  $(\hat{\mu}_1^{(n)}, \hat{\mu}_{-1}^{(n)}, \hat{\sigma}_1^{(n)}, \hat{\sigma}_{-1}^{(n)})$  converge as  $n \rightarrow \infty$  to the true parameter values with probability 1.*

*Proof.* The loglikelihood (70) is identical to that of a binary Gaussian mixture with known label marginals which we prove to be identifiable in Proposition 8. Consistency thus follows from classical MLE theory e.g., chapter 17 of [20].  $\square$

**Proposition 10.** *Under the assumptions of Proposition 8 and assuming the loss  $L$  is given by one of (61)-(63), the plug-in risk estimate (72) is consistent i.e.,  $\hat{R}_n(\theta) \rightarrow R(\theta)$  with probability 1.*

*Proof.* The plug-in risk estimate  $\hat{R}_n$  in (72) is a continuous function (when  $L$  is given by (61), (62) or (63)) of  $\hat{\mu}_1^{(n)}, \hat{\mu}_{-1}^{(n)}, \hat{\sigma}_1^{(n)}, \hat{\sigma}_{-1}^{(n)}$  (note that  $\mu_y$  and  $\sigma_y$  are functions of  $\theta$ ), which we denote  $\hat{R}_n(\theta) = h(\hat{\mu}_1^{(n)}, \hat{\mu}_{-1}^{(n)}, \hat{\sigma}_1^{(n)}, \hat{\sigma}_{-1}^{(n)})$ .

Using Proposition 9 we have that

$$\lim_{n \rightarrow \infty} (\hat{\mu}_1^{(n)}, \hat{\mu}_{-1}^{(n)}, \hat{\sigma}_1^{(n)}, \hat{\sigma}_{-1}^{(n)}) = (\mu_1^{\text{true}}, \mu_{-1}^{\text{true}}, \sigma_1^{\text{true}}, \sigma_{-1}^{\text{true}})$$

with probability 1. Since continuous functions preserve limits we have

$$\lim_{n \rightarrow \infty} h(\hat{\mu}_1, \hat{\mu}_2, \hat{\sigma}_1, \hat{\sigma}_2) = h(\mu_1^{\text{true}}, \mu_2^{\text{true}}, \sigma_1^{\text{true}}, \sigma_2^{\text{true}})$$

with probability 1 which implies convergence  $\lim_{n \rightarrow \infty} \hat{R}_n(\theta) = R(\theta)$  with probability 1.  $\square$

The above proposition shows that  $\hat{R}_n(\theta) \rightarrow R(\theta)$  with probability 1. Using standard arguments (Chapter 16 of [20]) it can be shown that for a compact  $\Theta$  we have the following uniform convergence  $\sup_\theta |\hat{R}_n(\theta) - R(\theta)| \rightarrow 0$  with probability 1 (under some

regularity conditions). Consequentially, the minimizer of the estimated risk converges to the minimum of the expected risk  $\arg \min_{\theta} \hat{R}_n(\theta) \rightarrow \arg \min_{\theta} R(\theta)$  (see e.g., Chapter 17 of [20]). The implication is profound: the above training procedure provides a classifier that converges to the optimal classifier (the minimizer of (60)) as the number of unlabeled examples increase without any labels whatsoever.

### 3.3 *Application 1: Estimating Risk in Transfer Learning*

We consider applying our estimation framework in two ways. The first application, which we describe in this section, is estimating margin-based risks in transfer learning where classifiers are trained on one domain but tested on a somewhat different domain. The transfer learning assumption that labeled data exists for the training domain but not for the test domain motivates the use of our unsupervised risk estimation. The second application, which we describe in the next section, is more ambitious. It is concerned with training classifiers without labeled data whatsoever.

In evaluating our framework we consider both synthetic and real-world data. In the synthetic experiments we generate high dimensional data from two uniform distributions  $X|Y = 1$  and  $X|Y = -1$  with independent dimensions and prescribed  $p(Y)$  and classification difficulty. This controlled setting allows us to examine the accuracy of the risk estimator as a function of  $n$ ,  $p(Y)$ , and the classifier accuracy.

Figure 20 (left and middle panels) shows that the relative error in estimating the logloss decreases with  $n$  achieving accuracy of greater than 99% for  $n > 1000$ . Interestingly, the figure shows that the estimation error decreases as the classifiers become more accurate and as  $p(Y)$  becomes less uniform. We found these trends to hold in other experiments as well. A hinge-loss estimation experiment resulted in similar results which we omit due to lack of space. In the case of exponential loss, however, the estimator performed substantially worse. This is likely due to the exponential dependency of the loss on  $Y f_{\theta}(X)$  which makes it very sensitive to outliers.

Data	$R_n$	$ R_n - \hat{R}_n $	$ R_n - \hat{R}_n /R_n$	$n$	$p(Y = 1)$
sci vs. comp	0.7088	0.0093	0.013	3590	0.8257
sci vs. rec	0.641	0.0141	0.022	3958	0.7484
talk vs. rec	0.5933	0.0159	0.026	3476	0.7126
talk vs. comp	0.4678	0.0119	0.025	3459	0.7161
talk vs. sci	0.5442	0.0241	0.044	3464	0.7151
comp vs. rec	0.4851	0.0049	0.010	4927	0.7972

**Figure 15:** Error in estimating logloss for logistic regression classifiers trained on one 20-newsgroup classification task and tested on another. We followed the transfer learning setup described in [16] which may be referred to for more detail. The train and test sets contained samples from two top categories in the topic hierarchy but with different subcategory proportions. As a result, the train and test distributions are similar but not identical. The first column indicates the top category classification task. The second column indicates the empirical log-loss  $R_n$  calculated using the true labels of the test set (66). The third and fourth columns indicate the absolute and the relative errors of the unsupervised logloss estimates. The fifth column  $n$  is the test set size and the last column is the label marginal  $p(y = 1)$ .

Figure 15 shows the accuracy of logloss estimation for a real world transfer learning experiment based on the 20-newsgroup data. Following the experimental setup of [16] we trained a classifier (logistic regression) on one 20 newsgroup classification problem and tested it on a related problem. Specifically, we used the hierarchical category structure to generate train and test sets with different distributions (see Figure 15 and [16] for more detail). The unsupervised estimation of the logloss risk was very effective with relative accuracy greater than 96% and absolute error less than 0.02.

### 3.4 Application 2: Unsupervised Learning of Classifiers

Our second application is a very ambitious one: training classifiers using only unlabeled data and  $p(Y)$ . We measure the performance of the learned classifier as a function of the unsupervised train set size  $n$ , in terms of  $\hat{R}_n$  (72) and in terms of the supervised logloss estimate  $R_n$  (66) (labels were used only in evaluation).

More specifically, we consider two algorithms (see Algorithms 1-2) that start with an initial  $\theta^{(0)}$  and iteratively construct a sequence of classifiers  $\theta^{(1)}, \dots, \theta^{(T)}$  which steadily improve the unsupervised logloss estimate (72)  $\hat{R}_n(\theta^{(t)}) \leq \hat{R}_n(\theta^{(t-1)})$ , as computed based

---

**Algorithm 1** Unsupervised Gradient Descent

---

**Input:**  $X^{(1)}, \dots, X^{(n)} \in \mathbb{R}^d, p(Y)$ , step size  $\alpha$

**repeat**

Initialize  $t = 0, \theta^{(t)} = \theta^0 \in \mathbb{R}^d$

Compute  $f_{\theta^{(t)}}(X^{(j)}) = \langle \theta^{(t)}, X^{(j)} \rangle \forall j = 1, \dots, n$

Estimate  $(\hat{\mu}_1, \hat{\mu}_{-1}, \hat{\sigma}_1, \hat{\sigma}_{-1})$  by maximizing (70)

**for**  $i = 1$  **to**  $d$  **do**

Plug-in the estimates into (72) to approximate

$$\frac{\partial \hat{R}_n(\theta^{(t)})}{\partial \theta_i} = \frac{\hat{R}_n(\theta^{(t)} + h_i e_i) - \hat{R}_n(\theta^{(t)} - h_i e_i)}{2h_i}$$

$(e_i \text{ is an all zero vector except for } [e_i]_i = 1)$  (74)

**end for**

Form  $\nabla \hat{R}_n(\theta^{(t)}) = \left( \frac{\partial \hat{R}_n(\theta^{(t)})}{\partial \theta_1^{(t)}}, \dots, \frac{\partial \hat{R}_n(\theta^{(t)})}{\partial \theta_d^{(t)}} \right)$

Update  $\theta^{(t+1)} = \theta^{(t)} - \alpha \nabla \hat{R}_n(\theta^{(t)}), t = t + 1$

**until** convergence

**Output:** linear classifier  $\theta^{\text{final}} = \theta^{(t)}$

---

---

**Algorithm 2** Unsupervised Grid Search

---

**Input:**  $X^{(1)}, \dots, X^{(n)} \in \mathbb{R}^d, p(Y)$ , grid-size  $\tau$

Initialize  $\theta_i \sim U(-2, 2)$  for all  $i$

**repeat**

**for**  $i = 1$  **to**  $d$  **do**

Construct  $\tau$  points grid in the range  $[\theta_i - 4\tau, \theta_i + 4\tau]$

Compute the risk estimate (72) where all dimensions of  $\theta^{(t)}$  are fixed except for  $[\theta^{(t)}]_i$  which is evaluated at each grid point.

Set  $[\theta^{(t+1)}]_i$  to the grid value that minimized (72)

**end for**

**until** convergence

**Output:** linear classifier  $\theta^{\text{final}} = \theta$

---

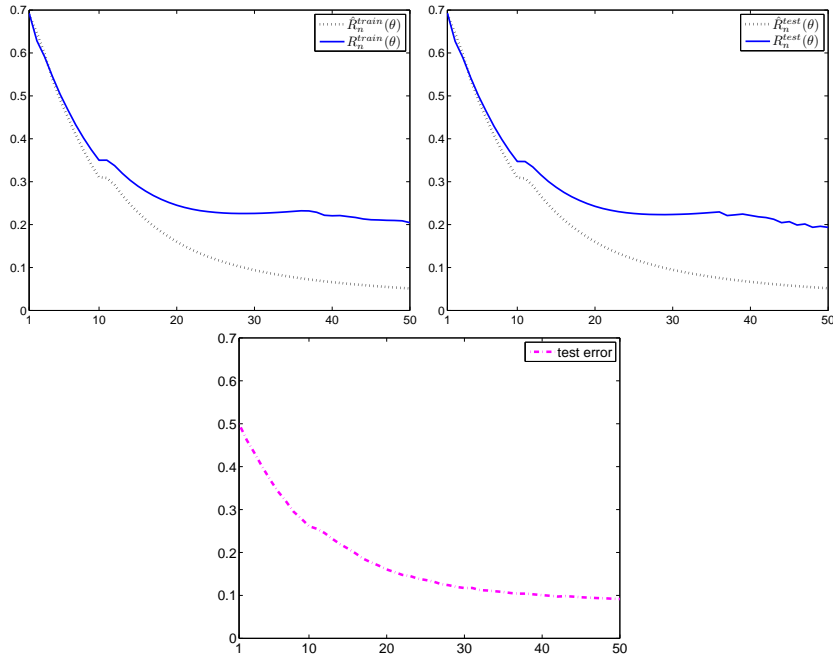
on an unlabeled training set of size  $n$ . Although we focus on unsupervised training of logistic regression (minimizing unsupervised logloss estimate), the same techniques may be generalized to train other margin-based classifiers such as SVM.

Algorithm 1 adopts a gradient descent-based optimization. At each iteration  $t$ , it approximates the gradient vector  $\nabla \hat{R}_n(\theta^{(t)})$  numerically using a finite difference approximation (74). Algorithm 2 proceeds by constructing a grid search along every dimension of  $\theta^{(t)}$  and set  $[\theta^{(t)}]_i$  to the grid value that minimizes  $\hat{R}_n$ .

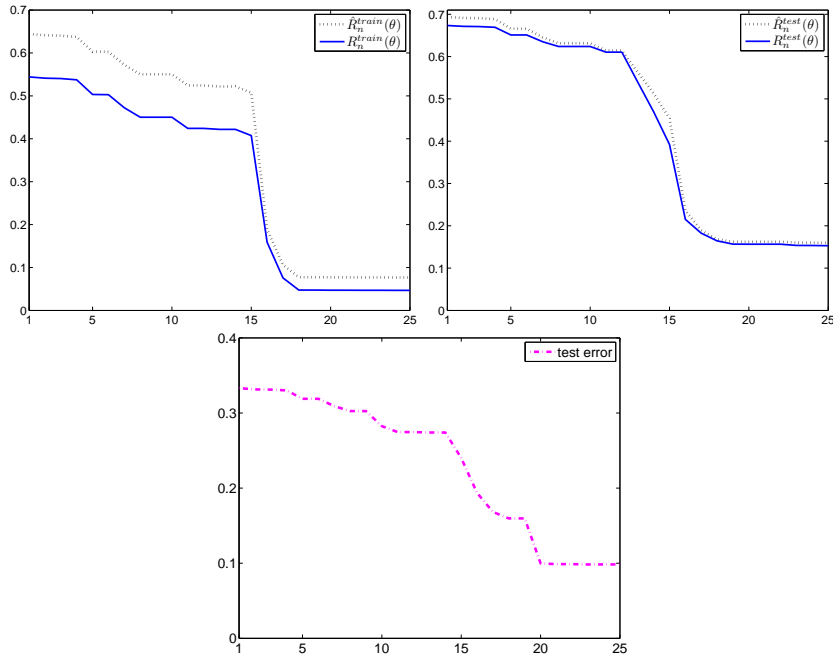
We tested the two algorithms on two real-world datasets: Reuters RCV1 text categorization and MNIST digit recognition datasets. In the case of RCV1 we discarded all but the most frequent 504 words (after stop-word removal) and represented documents using their tfidf scores. We experimented on the binary classification task of distinguishing the top category (positive) from the next 4 top categories (negative) which resulted in  $p(y = 1) = 0.3$  and  $n = 199328$ .

70% of the data was chosen as a (unlabeled) training set and the rest was held-out as a test-set. Figures 16-17 display the logloss estimates (both the unsupervised  $\hat{R}_n$  and the supervised  $R_n$ ) on the training and test sets as well as the test set error rate on RCV1 data. The classifiers were constructed by Algorithm 1 in Figure 16 and by Algorithm 2 in Figure 17.

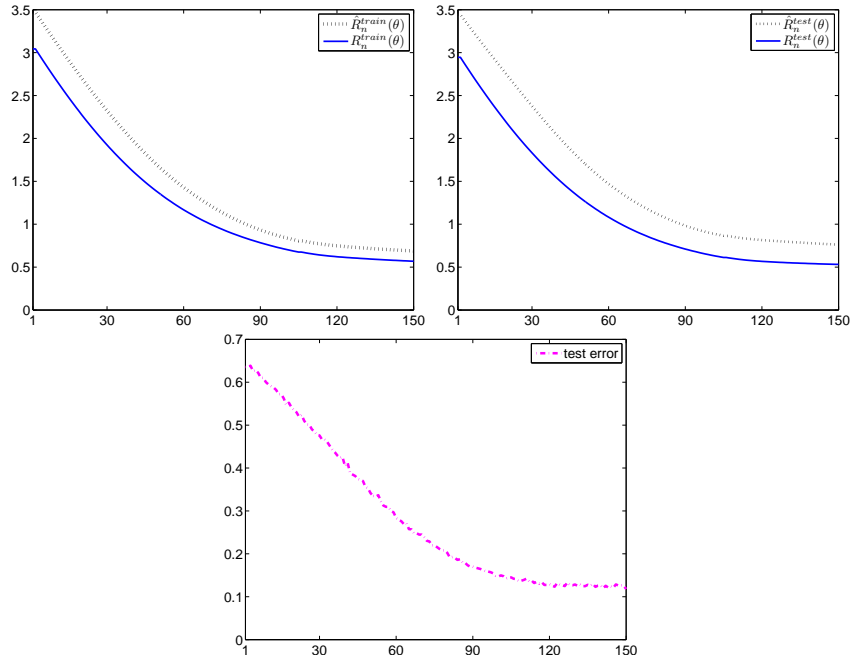
The results indicate that minimizing the unsupervised logloss estimate is quite effective in learning an accurate classifier without labels. Both algorithms reached test set error rate of 0.1 after 50 iterations (for gradient descent) and 25 iterations (for grid search). Training a supervised logistic regression on the same training set using labels yields test error rate of 0.07. This indicates that our approach achieves performance close to that of the supervised alternative without using a single label. Furthermore the two lines corresponding to the unsupervised  $\hat{R}_n$  and the supervised  $R_n$  decrease with  $t$  for both the train and test set. The improvement in accuracy and logloss was smoother for the gradient descent than for the grid search.



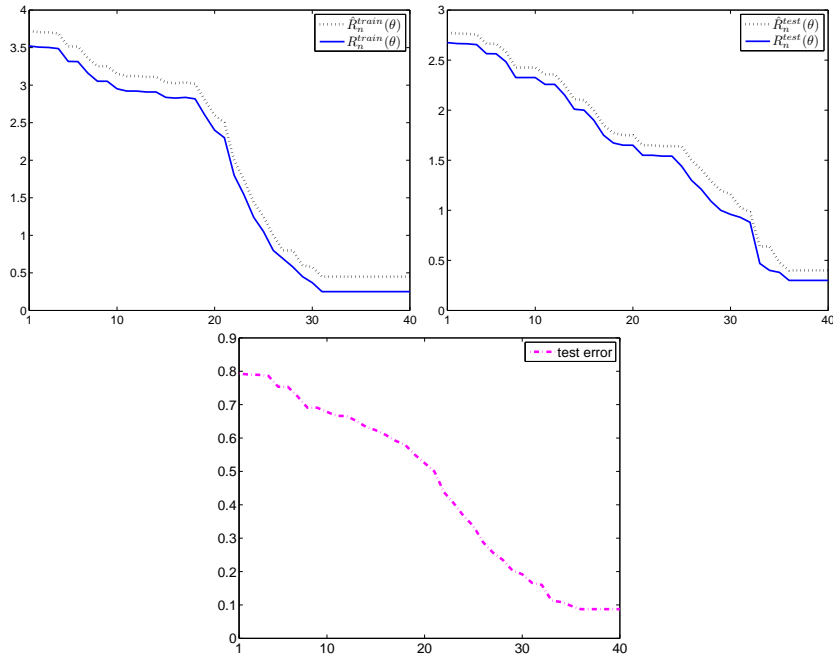
**Figure 16:** Estimation accuracy of classifiers learned by minimizing the unsupervised logloss estimate  $\hat{R}_n$  (72) on RCV1 data. The panels display the performance of the learned classifier in terms of the unsupervised  $\hat{R}_n$  and the supervised  $R_n$  logloss estimates based on the training set (left), based on the test set (middle) and the test classification error rate (right). The performance criteria are plotted as a function of the iteration number of Algorithm 1 (gradient descent). The figure shows that the algorithm obtains a relatively accurate classifier (test set error rate 0.1, and  $\hat{R}_n$  decaying similarly to  $R_n$ ) without the use of a single labeled example. The test error rate is 0.07 for supervised logistic regression. See text for more detail.



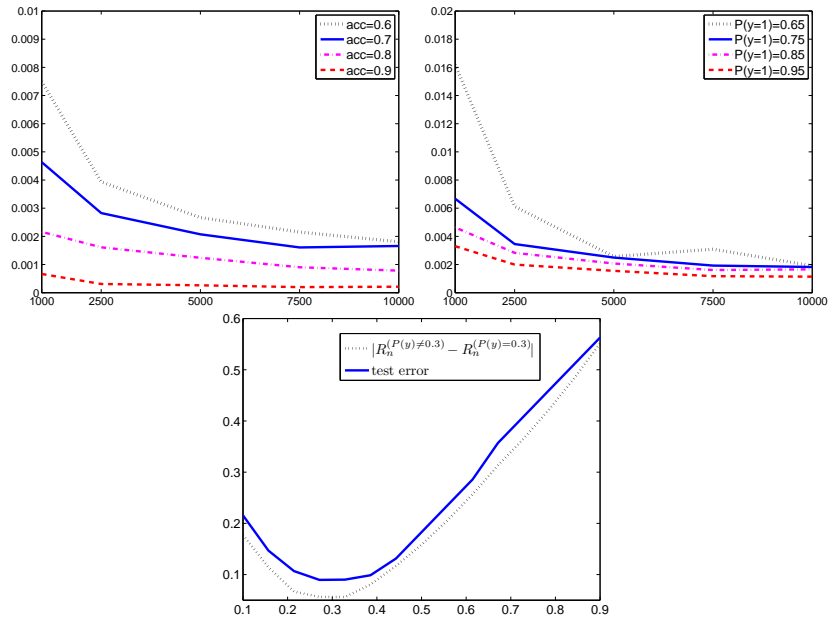
**Figure 17:** Estimation accuracy of classifiers learned by minimizing the unsupervised logloss estimate  $\hat{R}_n$  (72) on RCV1 data. The panels display the performance of the learned classifier in terms of the unsupervised  $\hat{R}_n$  and the supervised  $R_n$  logloss estimates based on the training set (left), based on the test set (middle) and the test classification error rate (right). The performance criteria are plotted as a function of the iteration number of Algorithm 2 (grid search). The figure shows that the algorithm obtains a relatively accurate classifier (test set error rate 0.1, and  $\hat{R}_n$  decaying similarly to  $R_n$ ) without the use of a single labeled example. The test error rate is 0.07 for supervised logistic regression. See text for more detail.



**Figure 18:** Estimation accuracy of classifiers learned by minimizing the unsupervised logloss estimate  $\hat{R}_n$  (72) on the MNIST data. The panels display the performance of the learned classifier in terms of the unsupervised  $\hat{R}_n$  and the supervised  $R_n$  logloss estimates based on the training set (left), based on the test set (middle) and the test classification error rate (right). The performance criteria are plotted as a function of the iteration number of Algorithm 1 (gradient descent). The figure shows that the algorithm obtains a relatively accurate classifier (test set error rate 0.1, and  $\hat{R}_n$  decaying similarly to  $R_n$ ) without the use of a single labeled example. The test error rate is 0.05 for supervised logistic regression. See text for more detail.



**Figure 19:** Estimation accuracy of classifiers learned by minimizing the unsupervised logloss estimate  $\hat{R}_n$  (72) on MNIST data. The panels display the performance of the learned classifier in terms of the unsupervised  $\hat{R}_n$  and the supervised  $R_n$  logloss estimates based on the training set (left), based on the test set (middle) and the test classification error rate (right). The performance criteria are plotted as a function of the iteration number of Algorithm 2 (grid search). The figure shows that the algorithm obtains a relatively accurate classifier (test set error rate 0.1, and  $\hat{R}_n$  decaying similarly to  $R_n$ ) without the use of a single labeled example. The test error rate is 0.05 for supervised logistic regression. See text for more detail.



**Figure 20:** Left and middle panels show the dependence of  $|\hat{R}_n - R_n|/R_n$  for logloss (based on synthetic data) on the number of unlabeled examples  $n$  and how it changes with the classifier accuracy (acc) and the label marginal  $p(Y)$ . The risk estimation error nicely decreases with  $n$  (approaching 1% relative error at  $n = 1000$  and decaying further). It also decreases with the accuracy of the classifier (left) and non-uniformity of  $p(Y)$ . Right panel: Performance of unsupervised classifier training on RCV1 data (top class vs. classes 2-5) for misspecified  $p(Y)$ . The performance of the estimated classifier (in terms of train set empirical logloss  $\hat{R}_n$  (66) and test error rate measured using held-out labels) decreases with the deviation between the assumed and true  $p(Y = 1)$  (true  $p(Y = 1) = 0.3$ ). The classifier performance is very good when the assumed  $p(Y)$  is close to the truth and degrades gracefully when the assumed  $p(Y)$  is not too far from the truth.

In the case of MNIST data, we normalized each of the  $28 \times 28 = 784$  pixels to have 0 mean and unit variance. Our classification task was to distinguish images of the digit one (positive) from the digit 2 (negative) resulting in 14867 samples and  $p(Y = 1) = 0.53$ . We randomly choose 70% of the data as a training set and kept the rest as a test set.

Figures 18 and 19 show the performance of the learned classifiers for the MNIST dataset for the gradient descent and the grid search algorithms. The results are similar to those obtained on the RCV1 dataset. The learned classifiers had test-set classification error rate of 0.1 and the decay of the train-set and test-set estimate  $\hat{R}_n$  as a function of the iteration number  $t$  closely mirrored the behavior of the supervised criterion  $R_n$ . For this data supervised logistic regression achieves test set error of 0.05 which is on the same order of magnitude as our unsupervised technique.

### 3.4.1 Inaccurate Specification of $p(Y)$

Our estimation framework assumes that the marginal  $p(Y)$  is known. In some cases we may only have an inaccurate estimate of  $p(Y)$ . It is instructive to consider how the performance of the learned classifier degrades with the inaccuracy of the assumed  $p(Y)$ .

Figure 20 (right) displays the performance of the learned classifier for RCV1 data as a function of the assumed value of  $p(Y = 1)$  (correct value is  $p(Y = 1) = 0.3$ ). We conclude that knowledge of  $p(Y)$  is an important component in our framework but precise knowledge is not crucial. Small deviations of the assumed  $p(Y)$  from the true  $p(Y)$  result in a small degradation of logloss estimation quality and test set error rate. Naturally, large deviation of the assumed  $p(Y)$  from the true  $p(Y)$  render the framework ineffective.

## CHAPTER IV

# ASYMPTOTIC ANALYSIS OF GENERATIVE SEMISUPERVISED LEARNING

### 4.1 Overview

Semisupervised learning (SSL) is a technique for estimating statistical models using both labeled and unlabeled data. The practical value of SSL has motivated several attempts to mathematically quantify its value beyond traditional supervised techniques.

Of particular importance is the dependency of that improvement on the amount of unlabeled and labeled data. In the case of structured prediction the accuracy of the SSL estimator depends also on the specific manner in which sequences are labeled. Focusing on the framework of generative or likelihood-based SSL applied to classification and structured prediction we identify the following questions which we address. *Q1: Consistency (classification)*. What combinations of labeled and unlabeled data lead to precise models in the limit of large data.

*Q2: Accuracy (classification)*. How can we quantitatively express the estimation accuracy for a particular generative model as a function of the amount of labeled and unlabeled data. What is the improvement in estimation accuracy resulting from replacing an unlabeled example with a labeled one.

*Q3: Consistency (structured prediction)*. What strategies for sequence labeling lead to precise models in the limit of large data.

*Q4: Accuracy (structured prediction)*. How can we quantitatively express the estimation quality for a particular model and structured labeling strategy. What is the improvement in estimation accuracy resulting from replacing one labeling strategy with another.

*Q5: Tradeoff (classification and structured prediction).* How can we quantitatively express the tradeoff between the two competing goals of improved prediction accuracy and low labeling cost. What are the possible ways to resolve that tradeoff optimally within a problem-specific context.

*Q6: Practical Algorithms.* How can we determine how much data to label in practical settings.

The first five questions are of fundamental importance to SSL theory. Recent related work has concentrated on large deviation bounds for discriminative SSL as a response to Q1 and Q2 above. While enjoying broad applicability, such non-parametric bounds are weakened when the model family's worst-case is atypical. By forgoing finite sample analysis, our approach complements these efforts and provides insights which apply to the specific generative models under consideration. In presenting answers to the last question, we reveal the relative merits of asymptotic analysis and how its employ, perhaps surprisingly, renders practical heuristics for controlling labeling cost.

Our asymptotic derivations are possible by extending the recently proposed stochastic composite likelihood formalism [17] and showing that generative SSL is a special case of that extension. The implications of this analysis are demonstrated using a simulation study as well as text classification and NLP structured prediction experiments. The developed framework, however, is general enough to apply to any generative SSL problem. As in [37], the delta method transforms our results from parameter asymptotics to prediction risk asymptotics.

## ***4.2 Related Work***

Semisupervised learning has received much attention in the past decade. Perhaps the first study in this area was done by Castelli and Cover [12] who examined the convergence of the classification error rate as a labeled example is added to an unlabeled dataset drawn from a Gaussian mixture model. Nigam et al. [40] proposed a practical SSL framework

based on maximizing the likelihood of the observed data. An edited volume describing more recent developments is [13].

The goal of theoretically quantifying the effect of SSL has recently gained increased attention. Sinha and Belkin [49] examined the effect of using unlabeled samples with imperfect models for mixture models. Balcan and Blum [3] and Singh et al. [48] analyze discriminative SSL using PAC theory and large deviation bounds. Additional analysis has been conducted under specific distributional assumptions such as the “cluster assumption”, “smoothness assumption” and the “low density assumption.”[13] However, many of these assumptions are criticized in [4].

Our work complements the above studies in that we focus on generative as opposed to discriminative SSL. In contrast to most other studies, we derive model specific asymptotics as opposed to non-parametric large deviation bounds. While such bounds are helpful as they apply to a broad set of cases, they also provide less information than model-based analysis due to their generality. Our analysis, on the other hand, requires knowledge of the specific model family and an estimate of the model parameter. The resulting asymptotics, however, apply specifically to the case at hand without the need of potentially loose bounds.

In particular, our work provides a new framework for examining the accuracy-cost SSL tradeoff in a way that is quantitative, practical, and model-specific.

### 4.3 *Stochastic SSL Estimators*

Generative SSL [40, 13] estimates a parametric model by maximizing the observed likelihood incorporating  $L$  labeled and  $U$  unlabeled examples

$$\ell(\theta) = \sum_{i=1}^L \log p_{\theta}(X^{(i)}, Y^{(i)}) + \sum_{i=L+1}^{L+U} \log p_{\theta}(X^{(i)}) \quad (75)$$

where  $p_{\theta}(X^{(i)})$  above is obtained by marginalizing the latent label  $\sum_y p_{\theta}(X^{(i)}, y)$ . A classical example is the naive Bayes model in [40] where  $p_{\theta}(X, Y) = p_{\theta}(X|Y)p(Y)$ ,  $p_{\theta}(X|Y = y) = \text{Mult}([\theta_y]_1, \dots, [\theta_y]_V)$ . The framework, however, is general enough to apply to any generative model  $p_{\theta}(X, Y)$ .

To analyze the asymptotic behavior of the maximizer of (75) we assume that the ratio between labeled to unlabeled examples  $\lambda = L/(L+U)$  is kept constant while  $n = L+U \rightarrow \infty$ . More generally, we assume a stochastic version of (75) where each one of the  $n$  samples  $X^{(1)}, \dots, X^{(n)}$  is labeled with probability  $\lambda$

$$\ell_n(\theta) = \sum_{i=1}^n Z^{(i)} \log p_\theta(X^{(i)}, Y^{(i)}) + \sum_{i=1}^n (1 - Z^{(i)}) \log p_\theta(X^{(i)}), \quad Z^{(i)} \sim \text{Bin}(1, \lambda). \quad (76)$$

The variable  $Z^{(i)}$  above is an indicator taking the value 1 with probability  $\lambda$  and 0 otherwise. Due to the law of large numbers for large  $n$  we will have approximately  $L = n\lambda$  labeled samples and  $U = n(1 - \lambda)$  unlabeled samples thus achieving the asymptotic behavior of (75).

Equation (32) is sufficient to handle the case of classification. However, in the case of structured prediction we may have sequences  $X^{(i)}, Y^{(i)}$  where for each  $i$  some components of the label sequence  $Y^{(i)}$  are missing and some are observed. For example one label sequence may be completely observed, another may be completely unobserved, and a third may have the first half labeled and the second half not.

More formally, we assume the existence of a sequence labeling policy or strategy  $\wp$  which maps label sequences  $Y^{(i)} = (Y_1^{(i)}, \dots, Y_m^{(i)})$  to a subset corresponding to the observed labels  $\wp(Y^{(i)}) \subset \{Y_1^{(i)}, \dots, Y_m^{(i)}\}$ . To achieve full generality we allow the labeling policy  $\wp$  to be stochastic, leading to different subsets of  $\{Y_1^{(i)}, \dots, Y_m^{(i)}\}$  with different probabilities. A simple “all or nothing” labeling policy could label the entire sequence with probability  $\lambda$  and otherwise ignore it. Another policy may label the entire sequence, the first half, or ignore it completely with equal probabilities

$$\wp(Y) = \begin{cases} Y_1^{(i)}, \dots, Y_m^{(i)} & \text{with probability } 1/3 \\ \emptyset & \text{with probability } 1/3 \\ Y_1^{(i)}, \dots, Y_{\lfloor m/2 \rfloor}^{(i)} & \text{with probability } 1/3 \end{cases} \quad (77)$$

We thus have the following generalization of (32) for structured prediction

$$\ell_n(\theta) = \sum_{i=1}^n \log p_{\theta}(\wp(Y^{(i)}), X^{(i)}). \quad (78)$$

Equation (78) generalizes standard SSL from all or nothing labeling to arbitrary labeling policies. The fundamental SSL question in this case is not simply what is the dependency of the estimation accuracy on  $n$  and  $\lambda$ . Rather we ask what is the dependency of the estimation accuracy on the labeling policy  $\wp$ . Of particular interest is the question what labeling policies  $\wp$  achieve high estimation accuracy coupled with low labeling cost. Answering these questions leads to a generative SSL theory that quantitatively balances estimation accuracy and labeling cost.

Finally, we note that both (32) and (78) are random variables whose outcomes depend on the random variables  $Z^{(1)}, \dots, Z^{(n)}$  (for (32)) or  $\wp$  (for (78)). Consequentially, the analysis of the maximizer  $\hat{\theta}_n$  of (32) or (78) needs to be done in a probabilistic manner.

#### 4.4 A1: Consistency (Classification)

Assuming that the data is generated from  $p_{\theta_0}(X, Y)$  consistency corresponds to the convergence of

$$\hat{\theta}_n = \arg \max_{\theta} \ell_n(\theta) \quad (79)$$

to  $\theta_0$  with probability 1 as  $n \rightarrow \infty$  ( $\ell_n$  is defined in (32)). This implies that in the limit of large data our estimator would converge to the truth. Note that large data  $n \rightarrow \infty$  in this case means that both labeled and unlabeled data increase to  $\infty$  (but their relative sizes remain the constant  $\lambda$ ).

We show in this section that the maximizer of (32) is consistent assuming that  $\lambda > 0$ . This is not an unexpected conclusion but for the sake of completeness we prove it here rigorously. The proof technique will also be used later when we discuss consistency of SSL estimators for structured prediction.

The central idea in the proof is to cast the generative SSL estimation problem as an extension of stochastic composite likelihood [17]. Our proof follows similar lines to the consistency proof of [17] with the exception that it does not assume independence of the indicator functions  $Z^{(i)}$  and  $(1 - Z^{(i)})$  as is assumed there.

**Definition 3.** A distribution  $p_\theta(X, Y)$  is said to be identifiable if  $\theta \neq \eta$  entails that  $p_\theta(X, Y) - p_\eta(X, Y)$  is not identically zero.

**Proposition 11.** Let  $\Theta \subset \mathbb{R}^r$  be a compact set, and  $p_\theta(x, y) > 0$  be identifiable and smooth in  $\theta$ . Then if  $\lambda > 0$  the maximizer  $\hat{\theta}_n$  of (32) is consistent i.e.,  $\hat{\theta}_n \rightarrow \theta_0$  as  $n \rightarrow \infty$  with probability 1.

*Proof.* The likelihood function, modified slightly by a linear combination with a constant is  $\ell'_n(\theta) =$

$$\begin{aligned} & \frac{1}{n} \sum_{i=1}^n (Z^{(i)} \log p_\theta(X^{(i)}, Y^{(i)}) - \lambda \log p_{\theta_0}(X^{(i)}, Y^{(i)})) + \\ & \frac{1}{n} \sum_{i=1}^n ((1 - Z^{(i)}) \log p_\theta(X^{(i)}) - (1 - \lambda) \log p_{\theta_0}(X^{(i)})), \end{aligned}$$

converges by the the strong law of large numbers as  $n \rightarrow \infty$  to its expectation with probability 1

$$\mu(\theta) = -\lambda D(p_{\theta_0}(X, Y) || p_\theta(X, Y)) - (1 - \lambda) D(p_{\theta_0}(X) || p_\theta(X)).$$

If we restrict ourselves to the compact set  $S = \{\theta : c_1 \leq \|\theta - \theta_0\| \leq c_2\}$  then  $|\log p_\theta(X, Y)| < K(X, Y) < \infty, \forall \theta \in S$ . As a result, the conditions for the uniform strong law of large numbers, cf. chapter 16 of [20], hold on  $S$  leading to

$$P \left\{ \lim_{n \rightarrow \infty} \sup_{\theta \in S} |\ell'_n(\theta) - \mu(\theta)| = 0 \right\} = 1. \quad (80)$$

Due to the identifiability of  $p_\theta(X, Y)$  we have  $D(p_{\theta_0}(X, Y) || p_\theta(X, Y)) \geq 0$  with equality iff  $\theta = \theta_0$ . Since also  $D(p_{\theta_0}(X) || p_\theta(X)) \geq 0$  we have that  $\mu(\theta) \leq 0$  with equality iff

$\theta = \theta_0$  (assuming  $\lambda > 0$ ). Furthermore, since the function  $\mu(\theta)$  is continuous it attains its negative supremum on the compact  $S$ :  $\sup_{\theta \in S} \mu(\theta) < 0$ .

Combining this fact with (80) we have that there exists  $N$  such that for all  $n > N$  the likelihood maximizers on  $S$  achieves strictly negative values of  $\ell'_n(\theta)$  with probability 1. However, since  $\ell'_n(\theta)$  can be made to achieve values arbitrarily close to zero under  $\theta = \theta_0$ , we have that  $\hat{\theta}_n \notin S$  for  $n > N$ . Since  $c_1, c_2$  were chosen arbitrarily  $\hat{\theta}_n \rightarrow \theta_0$  with probability 1.  $\square$

The above proposition is not surprising. As  $n \rightarrow \infty$  the number of labeled examples increase to  $\infty$  and thus it remains to ensure that adding an increasing number of unlabeled examples does not hurt the estimator. More interesting is the quantitative description of the accuracy of  $\hat{\theta}_n$  and its dependency on  $\theta_0, \lambda, n$  which we turn to next.

#### 4.5 A2: Accuracy (Classification)

The proposition below states that the distribution of the maximizer of (32) is asymptotically normal and provides its variance which may be used to characterize the accuracy of  $\hat{\theta}_n$  as a function of  $n, \theta_0, \lambda$ . As in Section 4.4 our proof proceeds by casting generative SSL as an extension of stochastic composite likelihood.

In Proposition 12 (below) and in Proposition 14 we use  $\text{Var}_{\theta_0}(H)$  to denote the variance matrix of a random vector  $H$  under  $p_{\theta_0}$ . The notations  $\xrightarrow{p}, \rightsquigarrow$  denote convergences in probability and in distribution [20] and  $\nabla f(\theta), \nabla^2 f(\theta)$  are the  $r \times 1$  gradient vector and  $r \times r$  matrix of second order derivatives of  $f(\theta)$ .

**Proposition 12.** *Under the assumptions of Proposition 11 as well as convexity of  $\Theta$  we have the following convergence in distribution of the maximizer of (32)*

$$\sqrt{n}(\hat{\theta}_n - \theta_0) \rightsquigarrow N(0, \Sigma^{-1}) \tag{81}$$

as  $n \rightarrow \infty$ , where

$$\Sigma = \lambda \mathbf{Var}_{\theta_0}(V_1) + (1 - \lambda) \mathbf{Var}_{\theta_0}(V_2)$$

$$V_1 = \nabla_{\theta} \log p_{\theta_0}(X, Y), \quad V_2 = \nabla_{\theta} \log p_{\theta_0}(X).$$

*Proof.* By the mean value theorem and convexity of  $\Theta$ , there is  $\eta \in (0, 1)$  for which  $\theta' = \theta_0 + \eta(\hat{\theta}_n - \theta_0)$  and

$$\nabla \ell_n(\hat{\theta}_n) = \nabla \ell_n(\theta_0) + \nabla^2 \ell_n(\theta')(\hat{\theta}_n - \theta_0).$$

Since  $\hat{\theta}_n$  maximizes  $\ell_n$  we have  $\nabla \ell_n(\hat{\theta}_n) = 0$  and

$$\sqrt{n}(\hat{\theta}_n - \theta_0) = -\sqrt{n} (\nabla^2 \ell_n(\theta'))^{-1} (\nabla \ell_n(\theta_0)). \quad (82)$$

By Proposition 11 we have  $\hat{\theta}_n \xrightarrow{p} \theta_0$  which implies that  $\theta' \xrightarrow{p} \theta_0$  as well. Furthermore, by the law of large numbers and the fact that  $W_n \xrightarrow{p} W$  implies  $g(W_n) \xrightarrow{p} g(W)$  for continuous  $g$ ,

$$\begin{aligned} (\nabla^2 \ell_n(\theta'))^{-1} &\xrightarrow{p} (\nabla^2 \ell_n(\theta_0))^{-1} \\ &\xrightarrow{p} \left( \lambda \mathbf{E}_{\theta_0} \nabla^2 \log p_{\theta_0}(X, Y) + (1 - \lambda) \mathbf{E}_{\theta_0} \nabla^2 \log p_{\theta_0}(X) \right)^{-1} = \Sigma^{-1} \end{aligned} \quad (83)$$

where in the last equality we used a well known identity concerning the Fisher information.

For the remaining term in the rhs of (82) we have

$$-\sqrt{n} \nabla \ell_n(\theta_0) = -\sqrt{n} \frac{1}{n} \sum_{i=1}^n (W^{(i)} + Q^{(i)}) \quad (84)$$

where  $W^{(i)} = Z^{(i)} \nabla \log p_{\theta_0}(X^{(i)}, Y^{(i)})$ ,  $Q^{(i)} = (1 - Z^{(i)}) \nabla \log p_{\theta_0}(X^{(i)})$ . Since (84) is an average of iid random vectors  $W^{(i)} + Q^{(i)}$  it is asymptotically normal by the central limit theorem with mean

$$\mathbf{E}_{\theta_0}(Q + W) = \lambda \mathbf{E}_{\theta_0} \nabla \log p_{\theta_0}(X, Y) + (1 - \lambda) \mathbf{E} \nabla \log p_{\theta_0}(X) = \lambda 0 + (1 - \lambda) 0.$$

and variance

$$\begin{aligned} \mathbf{Var}_{\theta_0}(W + Q) &= \mathbf{E}_{\theta_0} W^2 + \mathbf{E}_{\theta_0} Q^2 + 2 \mathbf{E}_{\theta_0} W Q \\ &= \lambda \mathbf{Var}_{\theta_0} V_1 + (1 - \lambda) \mathbf{Var}_{\theta_0} V_2 \end{aligned}$$

where we used  $\mathbf{E}(Z(1 - Z)) = \mathbf{E} Z - \mathbf{E} Z^2 = 0$ .

We have thus established that

$$-\sqrt{n}\nabla\ell_n(\theta_0) \rightsquigarrow N(0, \Sigma). \quad (85)$$

We finish the proof by combining (82), (89) and (85) using Slutsky's theorem.  $\square$

Proposition 12 characterizes the asymptotic estimation accuracy using the matrix  $\Sigma$ . Two convenient one dimensional summaries of the accuracy are the trace and the determinant of  $\Sigma$ . In some simple cases (such as binary event naive Bayes)  $\text{tr}(\Sigma)$  can be brought to a mathematically simple form which exposes its dependency on  $\theta_0, n, \lambda$ . In other cases the dependency may be obtained using numerical computing.

Figure 21 displays three error measures for the multinomial naive Bayes SSL classifier [40] and the Reuters RCV1 text classification data. In all three figures the error measures are represented as functions of  $n$  (horizontal axis) and  $\lambda$  (vertical axis). The error measures are classification error rate (left), trace of the empirical mse (middle), and log-trace of the asymptotic variance (right). The measures were obtained over held-out sets and averaged using cross validation. Figure 22 (middle) displays the asymptotic variance as a function of  $n$  and  $\lambda$  for a randomly drawn  $\theta_0$ .

As expected the measures decrease with  $n$  and  $\lambda$  in all the figures. It is interesting to note, however, that the shapes of the contour plots are very similar across the three different measures (top row). This confirms that the asymptotic variance (right) is a valid proxy for the finite sample measures of error rates and empirical mse. We thus conclude that the asymptotic variance is an attractive measure that is similar to finite sample error rate and at the same time has a convenient mathematical expression.

#### **4.6 A3: Consistency (Structured)**

In the case of structured prediction the log-likelihood (78) is specified using a stochastic labeling policy. In this section we consider the conditions on that policy that ensures estimation consistency, or in other word convergence of the maximizer of (78) to  $\theta_0$  as  $n \rightarrow \infty$ .

We assume that the labeling policy  $\wp$  is a probabilistic mixture of deterministic sequence labeling functions  $\chi_1, \dots, \chi_k$ . In other words,  $\wp(Y)$  takes values  $\chi_i(Y), i = 1, \dots, k$  with probabilities  $\lambda_1, \dots, \lambda_k$ . For example the policy (77) corresponds to  $\chi_1(Y) = Y$ ,  $\chi_2(Y) = \emptyset$ ,  $\chi_3(Y) = \{Y_1, \dots, Y_{\lfloor m/2 \rfloor}\}$  (where  $Y = \{Y_1, \dots, Y_m\}$ ) and  $\lambda = (1/3, 1/3, 1/3)$ .

Using the above notation we can write (78) as

$$\ell_n(\theta) = \sum_{i=1}^n \sum_{j=1}^k Z_j^{(i)} \log p_\theta(\chi_j(Y^{(i)}), X^{(i)}) \quad (86)$$

$$Z^{(i)} \sim \text{Mult}(1, (\lambda_1, \dots, \lambda_k))$$

which exposes its similarity to the stochastic composite likelihood function in [17]. Note however that (86) is not formally a stochastic composite likelihood since  $Z_j^{(i)}, j = 1, \dots, k$  are not independent and since  $\chi_j(Y)$  depends on the length of the sequence  $Y$  (see for example  $\chi_1$  and  $\chi_3$  above). We also use the notation  $S_j^m$  for the subset of labels provided by  $\chi_j$  on length- $m$  sequences

$$\chi_j(Y_1, \dots, Y_m) = \{Y_i : i \in S_j^m\}.$$

**Definition 4.** A labeling policy is said to be identifiable if the following map is injective

$$\bigcup_{m:q(m)>0} \bigcup_{j=1}^k \{p_\theta(\{Y_r : r \in S_j^m\}, X)\} \rightarrow p_\theta(X, Y)$$

where  $q$  is the distribution of sequences lengths. In other words, there is at most one collection of probabilities corresponding to the lhs above that does not contradict the joint distribution.

The importance of Definition 4 is that it ensures the recovery of  $\theta_0$  from the sequences partially labeled using the labeling policy. For example, a labeling policy characterized by  $\chi_1(Y) = Y_1$ ,  $\lambda_1 = 1$  (always label only the first sequence element) is non-identifiable for most interesting  $p_\theta$  as the first sequence component is unlikely to provide sufficient information to characterize the parameters associated with transitions  $Y_t \rightarrow Y_{t+1}$ .

**Proposition 13.** *Assuming the same conditions as Proposition 11, and  $\lambda_1, \dots, \lambda_k > 0$  with identifiable  $\chi_1, \dots, \chi_k$ , the maximizer of (86) is consistent i.e.,  $\hat{\theta}_n \rightarrow \theta_0$  as  $n \rightarrow \infty$  with probability 1.*

*Proof.* The log-likelihood (78), modified slightly by a linear combination with a constant is

$$\ell'_n(\theta) = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^k \left( Z_j^{(i)} \log p_\theta(\chi_j(Y^{(i)}), X^{(i)}) - \lambda_j \log p_{\theta_0}(\chi_j(Y^{(i)}), X^{(i)}) \right).$$

By the strong law of large numbers  $\ell'_n(\theta)$  converges to its expectation

$$\mu(\theta) = - \sum_{j=1}^k \lambda_j \sum_{m>0} q(m) \cdot D(p_{\theta_0}(\{Y_i : i \in S_j^m\}, X) || p_\theta(\{Y_i : i \in S_j^m\}, X)).$$

Since  $\mu$  is a linear combination of KL divergences with positive weights it is non-negative and is 0 if  $\theta = \theta_0$ . The identifiability of the labeling policy ensures that  $\mu(\theta) > 0$  if  $\theta \neq \theta_0$ . We have thus established that  $\ell'_n(\theta)$  converges to a non-negative continuous function  $\mu(\theta)$  whose maximum is achieved at  $\theta_0$ . The rest of the proof proceeds along similar lines as Proposition 13. □

Ultimately, the precise conditions for consistency will depend on the parametric family  $p_\theta$  under consideration. For many structured prediction models such as Markov random fields the consistency conditions are mild. Depending on the precise feature functions, consistency is generally satisfied for every policy that labels contiguous subsequences with positive probability. However, some care need to be applied for models like HMM containing parameters associated with the start label or end label and with models asserting higher order Markov assumptions.

#### 4.7 A4: Accuracy (Structured)

We consider in this section the dependency of the estimation accuracy in structured prediction SSL (78) on  $n, \theta_0$  but perhaps most interestingly on the labeling policy  $\wp$ . Doing so provides insight into not only how much data to label but also in what way.

**Proposition 14.** *Under the assumptions of Proposition 13 as well as convexity of  $\Theta$  we have the following convergence in distribution of the maximizer of (86)*

$$\sqrt{n}(\hat{\theta}_n - \theta_0) \rightsquigarrow N(0, \Sigma^{-1}) \quad (87)$$

as  $n \rightarrow \infty$ , where

$$\Sigma^{-1} = \mathbf{E}_{q(m)} \left\{ \sum_{j=1}^k \lambda_j \mathbf{Var}_{\theta_0}(\nabla V_{jm}) \right\}$$

$$V_{jm} = \log p_{\theta_0}(\{Y_i : i \in S_j^m\}, X).$$

*Proof.* By the mean value theorem and convexity of  $\Theta$  there is  $\eta \in (0, 1)$  for which  $\theta' = \theta_0 + \eta(\hat{\theta}_n - \theta_0)$  and

$$\nabla \ell_n(\hat{\theta}_n) = \nabla \ell_n(\theta_0) + \nabla^2 \ell_n(\theta')(\hat{\theta}_n - \theta_0).$$

Since  $\hat{\theta}_n$  maximizes  $\ell$ ,  $\nabla \ell_n(\hat{\theta}_n) = 0$  and

$$\sqrt{n}(\hat{\theta}_n - \theta_0) = -\sqrt{n}(\nabla^2 \ell_n(\theta'))^{-1} \nabla \ell_n(\theta_0). \quad (88)$$

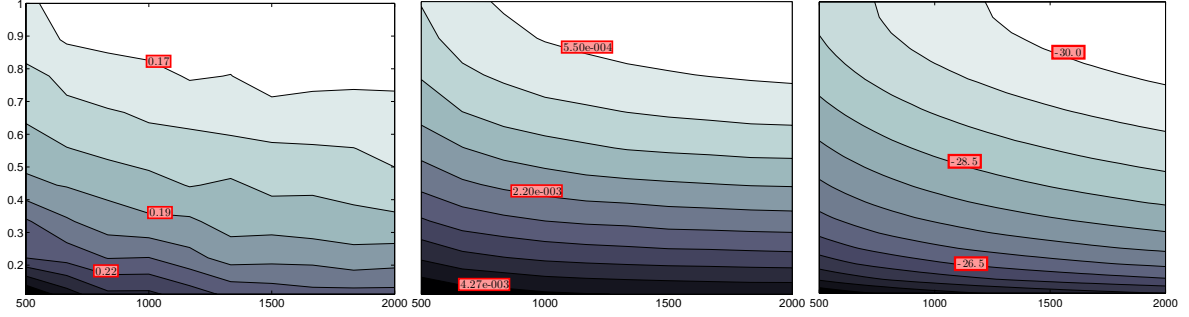
By Proposition 13 we have  $\hat{\theta}_n \xrightarrow{p} \theta_0$  which implies that  $\theta' \xrightarrow{p} \theta_0$  as well. Furthermore, by the law of large numbers and the fact that if  $W_n \xrightarrow{p} W$  then  $g(W_n) \xrightarrow{p} g(W)$  for continuous  $g$ ,

$$\begin{aligned} (\nabla^2 \ell_n(\theta'))^{-1} &\xrightarrow{p} (\nabla^2 \ell_n(\theta_0))^{-1} \\ &\xrightarrow{p} \left( \sum_{m>0} q(m) \sum_{j=1}^k \lambda_j \mathbf{E}_{\theta_0}(\nabla^2 V_{jm}) \right)^{-1} \\ &= - \left( \sum_{m>0} q(m) \sum_{j=1}^k \lambda_j \mathbf{Var}_{\theta_0}(\nabla V_{jm}) \right)^{-1}. \end{aligned} \quad (89)$$

where in the last equality we used a well known identity concerning the Fisher information.

For the remaining term on the rhs of (88) we have

$$\sqrt{n} \nabla \ell_n(\theta_0) = \sqrt{n} \frac{1}{n} \sum_{i=1}^n W_i \quad (90)$$



**Figure 21:** Three error measures for the multinomial naive Bayes SSL classifier applied to Reuters RCV1 text data. In each, error is a function of  $n$  (horizontal axis) and  $\lambda$  (vertical axis). The left depicts classification error rate, the middle depicts the trace of empirical mse, and right depicts the log-trace of the asymptotic variance. Results were obtained using held-out sets and averaged using cross validation. Particularly noteworthy is a striking correlation among all three figures, justifying the use of asymptotic variance as a surrogate for classification error, even for relatively small values of  $n$ .

where the random vectors

$$W_i = \sum_{m>0} 1_{\{\text{length}(Y^{(i)})=m\}} \sum_{j=1}^k Z_j^{(i)} \nabla V_{jm}^{(i)}$$

have expectation 0 due to the fact that the expectation of the score is 0. The variance of  $W_i$  is

$$\begin{aligned} \text{Var}_{\theta_0} W_i &= \mathbf{E}_{\theta_0} \sum_{m>0} 1_{\{\text{length}(Y^{(i)})=m\}} \sum_{j=1}^k Z_j^{(i)} \nabla V_{jm}^{(i)} \nabla V_{jm}^{(i)\top} \\ &= \sum_{m>0} q(m) \sum_{j=1}^k \lambda_j \mathbf{E} \left( \nabla V_{jm}^{(i)} \nabla V_{jm}^{(i)\top} \right) \end{aligned}$$

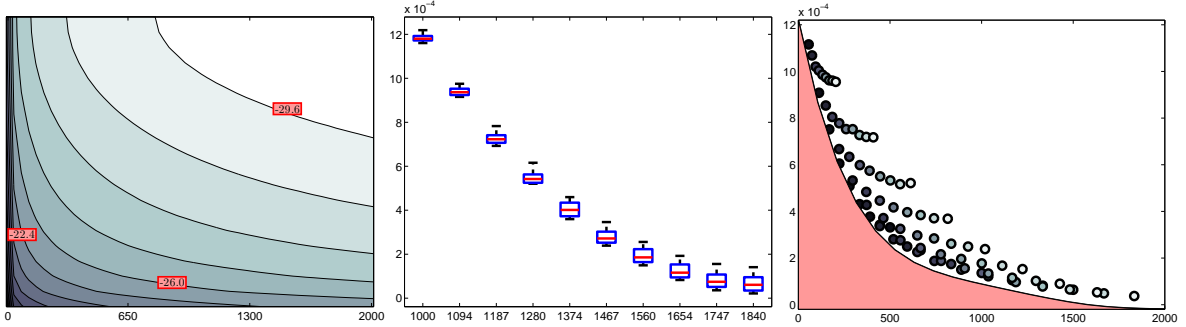
where in the first equality we used the fact that  $Y^{(i)}$  can have only one length and only one of  $\chi_1, \dots, \chi_k$  is chosen. Using the central limit theorem we thus conclude that

$$\sqrt{n} \nabla \ell_n(\theta_0) \rightsquigarrow N(0, \Sigma^{-1})$$

and finish the proof by combining (88), (89), and (85) using Slutsky's theorem.  $\square$

#### 4.8 A5: Tradeoff

As the figures in the previous sections display, the estimation accuracy increases with the total number of labels. The Cramer-Rao lower bound states that the highest accuracy is



**Figure 22:** Left figure represents log-trace of the theoretical variance and demonstrates phenomena under a simplified scenario, i.e., a mixture of two 1000-dim multinomials with unbalanced prior. Middle figure demonstrates the practical applicability of utilizing asymptotic analysis to characterize parameter error as a function of size of training-set partition. The training-set is fixed at 2000 samples and split for training and validating. As the proportion used for training is increased, we see a decrease in error. The shaded portion of the right panel depicts the empirically unachievable region for naive Bayes SSL classifier on the 20-newsgroups dataset.

obtained by the maximum likelihood operating on fully observed data. However, assuming that a certain cost is associated with labeling data SSL resolves a fundamental accuracy-cost tradeoff. A decrease in estimation accuracy is acceptable in return for decreased labeling cost.

Our ability to mathematically characterize the dependency of the estimation accuracy on the labeling cost leads to a new quantitative formulation of this tradeoff. Each labeling policy ( $\lambda, n$  in classification and  $\wp$  in structured prediction) is associated with a particular estimation accuracy via Propositions 12 and 14 and with a particular labeling cost. The precise way to measure labeling cost depends on the situation at hand, but we assume that the labeling cost is proportional to the numbers of labeled samples (classification) and of labeled sequence elements (structured prediction). This assumption may be easily relaxed by using other labeling cost functions e.g, obtaining unlabeled data may incur some cost as well.

Geometrically, each labeling policy may thus be represented in a two dimensional scatter plot where the horizontal and vertical coordinates correspond to labeling cost and estimation error respectively. The right panel in Figure 22 corresponds to multinomial naive

Bayes SSL classifier and the 20-newsgroups classification dataset. Each point in that panel corresponds to different  $n, \lambda$ .

The origin corresponds to the most desirable (albeit unachievable) position in the scatter plot representing zero error at no labeling cost. The cloud of points obtained by varying  $n, \lambda$  (classification) and  $\wp$  (structured prediction) represents the achievable region of the diagram. Most attractive is the lower and left boundary of that region which represents labeling policies that dominate others in both accuracy and labeling cost. The non-achievable region is below and to the left of that boundary (see shaded region in Figure 22, right). The precise position of the optimal policy on the boundary of the achievable region depends on the relative importance of minimizing estimation error and minimizing labeling cost. A policy that is optimal in one context may not be optimal in a different context. It is interesting to note that even in the case of naive Bayes classification (Figure 22, right) some labeling policies (corresponding to specific choices of  $n, \lambda$ ) are suboptimal. These policies correspond to points in the interior of the achievable region.

We consider in particular three different ways to define an optimal labeling policy (i.e., determining how much data to label) on the boundary of the achievable region

$$(\lambda^*, n^*)_1 = \arg \min_{(\lambda, n): \lambda n \leq C} \text{tr}(\Sigma^{-1}) \quad (91)$$

$$(\lambda^*, n^*)_2 = \arg \min_{(\lambda, n): \text{tr}(\Sigma^{-1}) \leq C} \lambda n \quad (92)$$

$$(\lambda^*, n^*)_3 = \arg \min_{(\lambda, n)} \lambda n + \alpha \text{tr}(\Sigma^{-1}). \quad (93)$$

The first applies in situations where the labeling cost is bounded by a certain available budget. The second applies when a certain estimation accuracy is acceptable and the goal is to minimize the labeling cost. The third considers a more symmetric treatment of the estimation accuracy and labeling cost.

Equations (91)-(93) may be easily generalized to arbitrary labeling costs  $f(n, \lambda)$ . Equations (91)-(93) may also be generalized to the case of structured prediction with  $\wp$  replacing  $(\lambda, n)$  and  $\text{cost}(\wp)$  replacing  $\lambda n$ .

## 4.9 A6: Practical Algorithms

Choosing a policy  $(\lambda, n)$  or  $\wp$  resolves the SSL tradeoff of accuracy vs. cost. Such a resolution is tantamount to answering the basic question of how many labels should be obtained (and in the case of structured prediction also which ones). Resolving the tradeoff via (91)-(93) or in any other way, or even simply evaluating the asymptotic accuracy  $\text{tr}(\Sigma)$  requires knowledge of the model parameter  $\theta_0$  that is generally unknown in practical settings.

We propose in this section a practical two stage algorithm for computing an estimate  $\hat{\theta}_n$  within a particular accuracy-cost tradeoff. Assuming we have  $n$  unlabeled examples, the algorithm begins the first stage by labeling  $r$  samples. It then estimates  $\theta'$  by maximizing the likelihood over the  $r$  labeled and  $n - r$  unlabeled samples. The estimate  $\hat{\theta}'$  is then used to obtain a plug-in estimate for the asymptotic accuracy  $\text{tr}(\Sigma)$ . In the second stage the algorithm uses the estimate  $\widehat{\text{tr}(\Sigma)}$  to resolve the tradeoff via (91)-(93) and determine how many more labels should be collected. Note that the labels obtained at the first stage may be used in the second stage as well with no adverse effect.

The two-stage algorithm spends some initial labeling cost in order to obtain an estimate for the quantitative tradeoff parameters. The final labeling cost, however, is determined in a principled way based on the relative importance of accuracy and labeling cost via (91)-(93). The selection of the initial number of labels  $r$  is important and should be chosen carefully. In particular it should not exceed the total desirable labeling cost.

We provide some experimental results on the performance of this algorithm in Figure 22 (middle). It displays box-plots for the differences between  $\text{tr}(\Sigma)$  and  $\widehat{\text{tr}(\Sigma)}$  as a function of the initial labeling cost  $r$  for naive Bayes SSL classifier and 20-newsgroups data. The figure illustrates that the two stage algorithm provides a very accurate estimation of  $\text{tr}(\Sigma)$  for  $r \geq 1000$  which becomes almost perfect for  $r \geq 1300$ .

## CHAPTER V

### FAST ALGORITHM FOR NONNEGATIVE TENSOR FACTORIZATION

In this chapter, we propose a fast algorithm for computing nonnegative tensor factorization, which can be subsequently used in many applications.

#### 5.1 Notations and operations

A tensor is a multi-dimensional array  $\mathbf{X} \in \mathbb{R}^{m_1 \times m_2 \times \dots \times m_N}$ . The order of a tensor is the number of dimensions, also known as way or mode. Mode- $n$  fiber of a given tensor is obtained by fixing every index except the  $n^{\text{th}}$  index.

**Definition 5.** An  $N$ -way tensor  $\mathbf{X} \in \mathbb{R}^{m_1 \times m_2 \times \dots \times m_N}$  is called a rank-1 tensor if it can be written as an outer product of  $N$  vectors  $a_1, \dots, a_N$ .

$$\mathbf{X} = a_1 \circ a_2 \circ a_3 \cdots \circ a_N$$

where  $\circ$  represents the vector outer product.

**Definition 6.** The Kronecker product of two matrices  $A \in \mathbb{R}^{I \times J}$  and  $B \in \mathbb{R}^{K \times L}$  is given by:

$$A \otimes B_{(IK) \times (JL)} = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1J}B \\ a_{21}B & a_{22}B & \cdots & a_{2J}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}B & a_{I2}B & \cdots & a_{IJ}B \end{pmatrix}$$

**Definition 7.** The Khatri-Rao product of two matrices  $A \in \mathbb{R}^{I \times J}$  and  $B \in \mathbb{R}^{K \times J}$  is given

by:

$$A \odot B_{(IK) \times (J)} = [a_1 \otimes b_1 \quad a_2 \otimes b_2 \quad \cdots \quad a_J \otimes b_J]$$

The process of flattening or unfolding a tensor is the reordering of the elements of the tensor in the form of a matrix. The mode- $n$  unfolding of a tensor  $\mathbf{X} \in \mathbb{R}_+^{m_1 \times m_2 \times \cdots \times m_N}$ , denoted by  $\mathbf{X}_{(n)}$ , is obtained by arranging the mode- $n$  fibers to be the columns of the resulting matrix. For example, consider a  $3 \times 3 \times 2$  tensor  $\mathbf{X} = (a_{ijk}) \in \mathbb{R}_+^{3 \times 3 \times 2}$ . The mode-1, 2, and 3 unfolded matrices will be, respectively,

$$\begin{aligned} X_{(1)}^T &= \begin{pmatrix} a_{1,1,1} & a_{1,2,1} & a_{1,3,1} & a_{1,1,2} & a_{1,2,2} & a_{1,3,2} \\ a_{2,1,1} & a_{2,2,1} & a_{2,3,1} & a_{2,1,2} & a_{2,2,2} & a_{2,3,2} \\ a_{3,1,1} & a_{3,2,1} & a_{3,3,1} & a_{3,1,2} & a_{3,2,2} & a_{3,3,2} \end{pmatrix} \\ X_{(2)}^T &= \begin{pmatrix} a_{1,1,1} & a_{2,1,1} & a_{3,1,1} & a_{1,1,2} & a_{2,1,2} & a_{3,1,2} \\ a_{1,2,1} & a_{2,2,1} & a_{3,2,1} & a_{1,2,2} & a_{2,2,2} & a_{3,2,2} \\ a_{1,3,1} & a_{2,3,1} & a_{3,3,1} & a_{1,3,2} & a_{2,3,2} & a_{3,3,2} \end{pmatrix} \\ X_{(3)}^T &= \begin{pmatrix} a_{1,1,1} & a_{1,2,1} & a_{1,3,1} & a_{2,1,1} & a_{2,2,1} & a_{2,3,1} & a_{3,1,1} & a_{3,2,1} & a_{3,3,1} \\ a_{1,1,2} & a_{1,2,2} & a_{1,3,2} & a_{2,1,2} & a_{2,2,2} & a_{2,3,2} & a_{3,1,2} & a_{3,2,2} & a_{3,3,2} \end{pmatrix}. \end{aligned}$$

### 5.1.1 Nonnegative tensor factorization

To discuss lower-rank factorization of tensors, one must define the rank of a tensor.

**Definition 8.** The rank of a tensor  $\mathbf{X}$  is defined as the minimum number of rank-one tensors needed, so that it can be represented in a polyadic form of those tensors.

A decomposition of a given tensor as a sum of rank-one tensors is called as PARAFAC decomposition. Even though the definition of tensor rank is similar to matrix rank, its properties are rather different. In fact, determining the rank of a given tensor is NP-complete [24]. An exact PARAFAC decomposition with  $r = \text{rank}(\mathbf{X})$  components is

called the rank decomposition of a tensor. An important property of tensor factorization is that the rank decompositions are unique up to elementary indeterminacies of scaling and permutation. It was claimed that the problem of computing PARAFAC decomposition of a tensor is well-posed for the case when it has nonnegative constraints [47]. We focus on nonnegative PARAFAC decomposition which we refer to as NTF.

Given an  $N$ -way nonnegative tensor,  $\mathbf{X} \in \mathbb{R}_+^{m_1 \times m_2 \times \dots \times m_N}$ , where  $\mathbb{R}_+$  denotes the positive quadrant, we need to decompose it into a set of loading matrices,  $\{A_1, A_2, \dots, A_N\}$ , where  $A_k \in \mathbb{R}_+^{m_k \times r}$  for  $1 \leq k \leq N$  and  $r$  is some positive integer<sup>1</sup>. For simplicity, we consider a three-way nonnegative tensor. Extension to higher order tensors is analogous.

**Definition 9.** Given a tensor  $\mathbf{X} = (x_{ijk}) \in \mathbb{R}_+^{m \times n \times p}$ , its NTF model with rank  $r$  is defined as

$$x_{ijz} = \sum_{q=1}^r a_{iq} b_{jq} c_{zq} + e_{ijz} \quad (94)$$

where  $A = a_{iq} \in \mathbb{R}_+^{m \times r}$ ,  $B = b_{jq} \in \mathbb{R}_+^{n \times r}$ ,  $C = c_{zq} \in \mathbb{R}_+^{p \times r}$ , are nonnegative loading matrices and  $\mathbf{E} = (e_{ijz})$  is the approximation error tensor.

## 5.2 NTF using ANLS framework and Block Principal Pivoting

### 5.2.1 ANLS framework

One of the problems with tensor decomposition is that there does not exist an algorithm for finding the exact number of components in the decomposition of a tensor [38]. An alternative way to proceed is to seek decompositions with multiple components and choose the best according to some criteria. For example, the loading matrices can be found by solving the following optimization problem

$$\min_{A, B, C \geq 0} \|\mathbf{X} - \llbracket ABC \rrbracket\|_F^2, \quad (95)$$

where  $\llbracket ABC \rrbracket = \sum_{q=1}^r a_q \circ b_q \circ c_q$  and is called a *Kruskal operator*.

---

<sup>1</sup>Various bounds exist in the literature for rank of a tensor. See, for example, [31] for further discussion on the rank of tensors.

This nonlinear optimization problem can be solved by using the alternating nonnegative least squares (ANLS) framework. For simplicity, we describe the framework for a three-way PARAFAC model with nonnegativity constraints although the framework can be extended to higher order models. In the case of a nonnegative tensor  $\mathbf{X} \in \mathbb{R}_+^{m \times n \times p}$ , we want to identify three nonnegative factors or loading matrices  $A \in \mathbb{R}_+^{m \times r}$ ,  $B \in \mathbb{R}_+^{n \times r}$ , and  $C \in \mathbb{R}_+^{p \times r}$ .

We first initialize two matrices, say  $B$  and  $C$ , and iterate solving the following nonnegativity constrained least squares (NNLS) problems until a stopping criteria is satisfied:

$$\min_{A \geq 0} \|Y_{BC}A^T - X_{(1)}\|_F^2 \quad (96)$$

$$\min_{B \geq 0} \|Y_{AC}B^T - X_{(2)}\|_F^2 \quad (97)$$

$$\min_{C \geq 0} \|Y_{AB}C^T - X_{(3)}\|_F^2 \quad (98)$$

where  $Y_{BC} = B \odot C$  and  $X_{(1)}$  is the  $(np) \times m$  unfolded matrix;  $Y_{AC} = A \odot C$  and  $X_{(2)}$  is the  $(mp) \times n$  unfolded matrix, and  $Y_{AB} = A \odot B$  and  $X_{(3)}$  is the  $(mn) \times p$  unfolded matrix.

This ANLS formulation has the property that for any  $N > 2$ , if each of the sub problems have unique solution, then the limit point of the sequence is a stationary point [7]. For  $N = 2$ , any limit point of the sequence is a stationary point [39]. The unfolding operation is critical in the sense that it turns the original problem into a sequence of NNLS problems. Another important thing to note is the structure of the NNLS problems that are obtained by the unfolding operation. Typically,  $r$  is small for a low rank approximation, and hence the matrices formed by the Khatri-rao product is long and thin. This observation is important in designing efficient algorithms for solving the NNLS problems. In the following section, we explain how we efficiently solve the NNLS problems.

---

**Algorithm 3** NTF/ANLS
 

---

- Given a tensor  $\mathbf{X} \in \mathbb{R}_+^{m_1 \times m_2 \times \dots \times m_N}$  and rank  $r$ , initialize the  $N - 1$  of the loading matrices, say  $A_2 \in \mathbb{R}_+^{m_2 \times r} \dots A_N \in \mathbb{R}_+^{m_N \times r}$  with nonnegative values.
- Repeat solving the following NNLS problems using the block principal pivoting method until a stopping criterion is satisfied:

$$\begin{aligned}
 & \min_{A_1 \geq 0} \|Y_{A_2 \dots A_N} A_1^T - X_{(1)}\|_F^2 \\
 & \min_{A_2 \geq 0} \|Y_{A_1 A_3 \dots A_N} A_2^T - X_{(2)}\|_F^2 \\
 & \quad \vdots \\
 & \min_{A_N \geq 0} \|Y_{A_1 \dots A_{N-1}} A_N^T - X_{(N)}\|_F^2
 \end{aligned}$$


---

### 5.2.2 Block Principal Pivoting Algorithm

The computational task for NTF is now narrowed down to the NNLS problems in Eqs. (96)-(98). For the moment, suppose we want to solve a NNLS problem given as

$$\min_{X \geq 0} \|DX - E\|_F^2, \tag{99}$$

where  $D \in \mathbb{R}^{p \times q}$ ,  $E \in \mathbb{R}^{p \times l}$ , and  $X \in \mathbb{R}^{q \times l}$ . Note that one can solve Eq. (99) by naively solving NNLS problems for each right-hand side vector, which appears as

$$\min_{x \geq 0} \|Dx - e\|_2^2. \tag{100}$$

Although this approach is possible, we will see that there exist efficient ways to accelerate the multiple right-hand side case.

Algorithms for solving Eq. (99) or (100) have been studied by others [11, 5, 29]. For each case of applications, the algorithm of choice depends on the size and the structure of NNLS problems. Note that in the case of NTF, because  $D$  is typically long and thin after the unfolding operation, each column of  $X$  is rather short. In fact, the size of each column vector is the target lower dimension,  $r$ . Hence, active-set-type methods are expected to perform well compared to iterative optimization schemes. We adopted the modified

active-set-type algorithm, called the block principal pivoting method, studied by Kim and Park [29]. They efficiently extended the block principal pivoting algorithm in [43] for the multiple right-hand side case. Here we briefly summarize the key ideas of [43] and [29].

The motivation of block principal pivoting methods [43] comes from the difficulty of conventional active set algorithms which occur when the number of variables increases. In active set algorithms, because typically only one variable is exchanged per iteration between the active and passive sets, the number of iterations heavily depend on number of variables. To accelerate computation, an algorithm whose iteration count does not depend on the number of variables is desirable. The block principal pivoting methods manage to do so by exchanging multiple variables at a time. To describe the method, let us consider the NNLS problem with a single right-hand side in Eq. (100). The Karush-Kuhn-Tucker (KKT) optimality conditions for Eq. (100) are given as

$$y = D^T D x - D^T e, \quad (101a)$$

$$y \geq 0, \quad x \geq 0, \quad (101b)$$

$$x_i y_i = 0, \quad i = 1, \dots, q. \quad (101c)$$

We assume that the matrix  $D$  has full column rank. In this case, a solution  $x$  that satisfies the conditions in Eqs. (101) is the optimal solution of Eq. (100).

We divide the index set  $\{1, \dots, q\}$  into two subgroups  $F$  and  $G$  where  $F \cup G = \{1, \dots, q\}$  and  $F \cap G = \phi$ . Let  $x_F$ ,  $x_G$ ,  $y_F$ , and  $y_G$  denote the subsets of variables with corresponding indices, and let  $D_F$  and  $D_G$  denote the submatrices of  $D$  with corresponding column indices. Initially, we assign  $x_G = 0$  and  $y_F = 0$ . Then, by construction,  $x = (x_F, x_G)$  and  $y = (y_F, y_G)$  always satisfy Eq. (101c) for any  $x_F$  and  $y_G$ . Now, we compute  $x_F$  and  $y_G$  using Eq. (101a) and check whether the computed values of  $x_F$  and  $y_G$  satisfy Eq. (101b). Computation of  $x_F$  and  $y_G$  is done as follows:

$$D_F^T D_F x_F = D_F^T e, \quad (102a)$$

$$y_G = D_G^T (D_F x_F - e). \quad (102b)$$

One can first solve for  $x_F$  in Eq. (102a) and use it to compute  $y_G$  in Eq. (102b). We call the computed pair  $(x_F, y_G)$  a complementary basic solution.

If a complementary basic solution  $(x_F, y_G)$  satisfies  $x_F \geq 0$  and  $y_G \geq 0$ , then it is called *feasible*. In this case,  $x = (x_F, 0)$  is the optimal solution of Eq. (100), and the algorithm terminates. Otherwise, a complementary basic solution  $(x_F, y_G)$  is *infeasible*, and we need to update  $F$  and  $G$  by exchanging variables for which Eq. (101b) or Eq. (101c) does not hold. Formally, we define the following index sets

$$H_1 = \{i \in F : x_i < 0\} \quad (103a)$$

$$H_2 = \{i \in G : y_i < 0\}, \quad (103b)$$

and update  $F$  and  $G$  by the following rules:

$$F = (F - H_1) \cup H_2 \quad (104a)$$

$$G = (G - H_2) \cup H_1. \quad (104b)$$

The finite termination property of this strategy with careful modifications is discussed in [43].

For a multiple right-hand side case in Eq. (99), Kim and Park [29] significantly improved this algorithm by employing two important improvements. Observe that the sets  $F$  and  $G$  change over iterations, and Eqs. (102) has to be solved for different  $F$  and  $G$  every time. The first improvement is based on the observation that the matrix  $D$  is typically very long and thin. In this case, constructing matrices  $D_F^T D_F$ ,  $D_F^T e$ ,  $D_G^T D_F$ , and  $D_G^T e$  before solving Eqs. (102) is computationally very expensive. To ease this difficulty,  $D^T D$  and  $D^T E$  can be computed in the beginning and reused in later iterations. One can easily see that  $D_F^T D_F$ ,  $D_F^T e_j$ ,  $D_G^T D_F$ , and  $D_G^T e_j$ ,  $j \in \{1, \dots, l\}$ , can be directly retrieved as a submatrix of  $D^T D$  and  $D^T E$ . Because the column size of  $C$  is small, storage needed for  $D^T D$  and  $D^T E$  is also small.

The second improvement involves exploiting common computations in solving Eq. 102a. Here we simultaneously run the block principal pivoting approach mentioned above

for multiple right-hand side vectors. At each iteration, we have the index sets  $F_j$  and  $G_j$  for each column  $j \in \{1, \dots, l\}$ , and we must compute  $x_{F_j}$  and  $y_{G_j}$  using Eqs. (102). The idea is to find groups of columns that share the same index sets  $F_j$  and  $G_j$ . We reorder the columns with respect to these groups and solve Eqs. (102) for the columns in the same group. By doing so, we avoid repeated Cholesky factorization computations required for solving Eq. (102a). With these modifications, the block principal pivoting method appeared very efficient in solving NNLS problems with multiple right hand sides [29].

### 5.3 Extensions to Regularized and Sparse NTF

In this section, we show how the algorithm can be extended to regularized and sparse NTF. There are multiple interpretations for regularization. It aids in numerical stability by solving a different but more stable problem. On the other hand, from a Bayesian viewpoint, regularization is a maximum a-posteriori (MAP) approximation, and it enforces prior beliefs provided by domain experts.

For regularized NTF, the objective function to be minimized is

$$\begin{aligned} \min_{A,B,C \geq 0} & \|\mathbf{X} - \llbracket ABC \rrbracket\|_F^2 + \alpha \|A\|_F^2 \\ & + \beta \|B\|_F^2 + \gamma \|C\|_F^2. \end{aligned} \quad (105)$$

Here we iterate solving the following NNLS problems until a stopping criterion is satisfied:

$$\begin{aligned} \min_{A \geq 0} & \left\| \begin{pmatrix} Y_{BC} \\ \sqrt{\alpha} I_{r \times r} \end{pmatrix} A^T - \begin{pmatrix} \mathbf{X}^{(1)} \\ 0_{r \times m} \end{pmatrix} \right\|_F^2, \\ \min_{B \geq 0} & \left\| \begin{pmatrix} Y_{AC} \\ \sqrt{\beta} I_{r \times r} \end{pmatrix} B^T - \begin{pmatrix} \mathbf{X}^{(2)} \\ 0_{r \times n} \end{pmatrix} \right\|_F^2, \\ \min_{C \geq 0} & \left\| \begin{pmatrix} Y_{AB} \\ \sqrt{\gamma} I_{r \times r} \end{pmatrix} C^T - \begin{pmatrix} \mathbf{X}^{(3)} \\ 0_{r \times p} \end{pmatrix} \right\|_F^2, \end{aligned}$$

where  $\alpha, \beta, \gamma$  are regularization coefficients,  $I_{r \times r}$  is an  $r \times r$  identity matrix, and  $0_{x \times y}$  is a zero matrix of dimensions  $x \times y$ . The role of the parameters  $\alpha, \beta, \gamma$  with small values is

to impose full rank on the matrices on the left hand side of variable matrices in the NNLS subproblems.

Our sparse NTF formulation incorporates  $\ell_1$ -norm regularization. The idea of using  $\ell_1$ -norm regularization for the purpose of achieving sparsity has been successfully utilized in a variety of problems [55]. Without loss of generality, we assume that one of the loading matrix, say  $A$ , is to be sparse. In that case, sparse NTF can be viewed as minimizing the following objective function:

$$\begin{aligned} \min_{A,B,C \geq 0} & \|\mathbf{X} - \llbracket ABC \rrbracket\|_F^2 + \alpha \sum_{j=1}^r \|A(:, j)\|_1^2 \\ & + \beta \|B\|_F^2 + \gamma \|C\|_F^2. \end{aligned} \quad (106)$$

In this case, we iterate solving the following NNLS problems until a stopping criterion is satisfied:

$$\begin{aligned} \min_{A \geq 0} & \left\| \begin{pmatrix} Y_{BC} \\ \sqrt{\alpha} e_{1 \times r} \end{pmatrix} A^T - \begin{pmatrix} \mathbf{X}^{(1)} \\ 0_{1 \times m} \end{pmatrix} \right\|_F^2, \\ \min_{B \geq 0} & \left\| \begin{pmatrix} Y_{AC} \\ \sqrt{\beta} I_{r \times r} \end{pmatrix} B^T - \begin{pmatrix} \mathbf{X}^{(2)} \\ 0_{r \times n} \end{pmatrix} \right\|_F^2, \\ \min_{C \geq 0} & \left\| \begin{pmatrix} Y_{AB} \\ \sqrt{\gamma} I_{r \times r} \end{pmatrix} C^T - \begin{pmatrix} \mathbf{X}^{(3)} \\ 0_{r \times p} \end{pmatrix} \right\|_F^2, \end{aligned}$$

where  $\alpha$  is a coefficient controlling sparsity,  $e_{1 \times r}$  is a vector of ones.

## 5.4 Experiments and Results

We experimented with various data sets from different application domains. The data sets and their size are shown in Table 1, and each of the data sets are explained in more detail below. We compared the following algorithms for NTF:

1. **(BPP)** Proposed NTF using the block principal pivoting method

Data set	Dimensions	Speed up
Synthetic	$100 \times 433 \times 200$	3.6
Amino-acid	$5 \times 201 \times 61$	7.4
CMU face images	$128 \times 120 \times 640$	9.8
Enron (3-way)	$3000 \times 141 \times 141$	12.2
Enron (4-way)	$39573 \times 197 \times 197 \times 357$	11.3
IEEE Vast 2007	$12121 \times 7141 \times 15$	8.9

**Table 1:** Data sets used, their sizes, and amount of speed-up: The values in “Speed up” column represent the amount of average speedup the **BPP** algorithm has over the second fastest algorithm.

2. (**ACTSET**) NTF using the active set method [28]
3. (**AB**) Andersson and Bro’s NTF [1]
4. (**MU**) NTF using multiplicative updates [35, 58]

For fair timing comparison, we present the relative residual as a function of time to see the convergence and the rate of convergence of each algorithm. Below, we give a brief description of each data set used, and we proceed to timing and application results. All the experiments were performed with MATLAB version 7.8 in a 3.2 GHZ Pentium 4 machine with Linux OS.

#### 5.4.1 Description of data sets

We used data sets from various domains including text mining, social network (email network) analysis, image processing, and bioinformatics. This illustrates the wide applicability of the proposed NTF algorithm.

*Synthetic tensor:* For this data set, multiple tensors of size  $100 \times 433 \times 200$  were generated. Each entry of the tensor was sampled independently from a uniform distribution  $U(0, 1)$  making the random tensor inherently nonnegative.

*Amino acid data set*<sup>2</sup>: This data set consists of fluorescence data (AMINO) containing five samples with different amounts of tryptophan, phenylalanine, and tyrosine. This data set has a small number of negative values which came from the intrinsic uncertainty in real

---

<sup>2</sup><http://www.models.life.ku.dk/research/data/AminoAcid/fluo/index.asp>

experimental measurements that included noise. Hence, having such small negative values is not contradictory to the nonnegativity assumption; in other words, true parameters are still nonnegative. It is a relatively small 3-way data set, with dimensionality  $5 \times 201 \times 61$ . Each sample was excited at 61 wavelengths (240 - 300 nm in 1 nm interval), and fluorescence emission intensities are measured at 201 wavelengths (250 - 450 nm in 1 nm interval). Each element of the tensor represents fluorescence emission signal intensity. A three-component PARAFAC model ( $k = 3$ ) was chosen for this data set since we already know that each signal intensity comes from three analytes. The values in the first, second, and third loading matrices represent the sample mode loadings, the emission mode loadings, and the excitation mode loadings, respectively.

*CMU Face data set*<sup>3</sup>: This data set consists of 640 facial images taken with varying pose, expression, eyes, and size. Each image is of dimension  $128 \times 120$ . Standard image processing approaches treat each image as a 15,360 dimensional vector, but tensors provide a way to model images without the need for vectorizing them, thereby preserving the inherent spatial relationship in the image. In addition, all the pixel values are inherently non-negative, and hence we impose the non-negativity constraint on the factors.

*Enron email data set*<sup>4</sup>: The Enron email data set consists of email exchanges between employees of Enron corporation. The raw Enron corpus contains 619,446 messages belonging to 158 users. For experimental purposes, a subset of Enron emails was selected. We created two types of tensor data sets from the raw data set. The base directory in both cases consisted of 121,393 terms.

For the first experiment, a 4-way term-author-recipient-day tensor with  $39,573 \times 197 \times 197 \times 357$  dimensions was constructed. This can be useful for tracking discussion between the users on a day-by-day level about particular topics. The  $ijkl^{th}$  element of the tensor represents the count of the term  $i$  used by author  $j$  while emailing recipient  $k$  on the  $l^{th}$  day.

---

<sup>3</sup><http://mlr.cs.umass.edu/ml/datasets/CMU+Face+Images>

<sup>4</sup><http://www.cs.cmu.edu/~enron/>

For next experiment, we created a 3-way author-recipient-keyword tensor with  $3,000 \times 141 \times 141$  dimensions. This tensor did not take into account the time stamp of the emails. The  $ijk^{th}$  entry represents how many times the user  $i^{th}$  had used the  $k^{th}$  keyword, while conversing with user  $j$ . This tensor was created to see which group of people talked to each other about what particular topic.

*IEEE VAST 2007 Contest data set*<sup>5</sup>: This data set consists of a tensor formed out of 1,455 text files corresponding to news stories, email messages, or blog posts from the VAST data set. In addition to the plain text versions of these files, the data set includes tag information such as date, person, location, organization, and money. The date tag was used to help extract the time stamp. The four remaining tags all represent entities of interest and were used to create the tensor model. We considered term-by-entity associations in the news stories over monthly time intervals, which corresponds to a sparse tensor in  $12, 121 \times 7, 141 \times 15$  dimensions with 1, 142, 077 nonzeros. Detailed expressions for calculating the entries of the tensor can be found in [2].

### 5.4.2 Timing comparison

The algorithms were compared for speed and the relative sum of squares residual error (RSSR):

$$RSSR = \frac{\sum_{i,j,z} e_{ijz}^2}{\sum_{i,j,z} x_{i,j,z}^2}, \quad (107)$$

where  $e_{ijz}$  and  $x_{i,j,z}$  are as shown in Eq. (94). We denote the  $RSSR$  value of  $t$ -th iteration by  $RSSR(t)$ .

For Enron, VAST, CMU face, and synthetic data sets, the results are reported in Figure 24. The graphs show the  $RSSR$  values with respect to computation time for several algorithms. From the figures, it is clear that our new method outperforms other existing algorithms: Regardless of the duration that the algorithms are run and stopped, the **BPP** method would provide the lowest  $RSSR$  value. The advantage of block principal pivoting

---

<sup>5</sup><http://www.cs.umd.edu/hcil/VASTcontest07>

method is generally greater when the number of factors to be recovered ( $r$ ) is larger. For the amino acid data set, the results are reported in Table 2. The table shows the  $RSSR$  values achieved at the specified amount of time. Note that the **BPP** method achieved a comparable  $RSSR$  value at a smaller amount of time.

Results for regularized and sparse NTF are reported in Table 3 for **BPP** and **ACTSET** algorithms. The timing values in the table are calculated as follows. The **BPP** algorithm was run until  $(RSSR(t-1) - RSSR(t))$  was less than  $10^{-6}$ , and the value of  $RSSR$  at the iteration  $t$  was noted. The corresponding time was reported for **BPP** algorithm. Next, other algorithms were run until they achieve same value of  $RSSR$ . **ACTSET** algorithm achieved the  $RSSR$  value at the time mentioned, which is greater than the timing reported for **BPP**. Other algorithms took longer than **ACTSET** algorithm.

To summarize the relative efficiency, we report in Table 1 the average speedup of our algorithm over the second fastest algorithm. We first picked 30  $RSSR$  values randomly from a uniform distribution  $U(RSSR_{min}, 1)$  where  $RSSR_{min}$  was the smallest observed  $RSSR$  value for each data set. Then, we measured time required to achieve each of the  $RSSR$  values by each algorithm. For each case, we calculated the speedup of the **BPP** algorithm over the second fastest one, and the average speedup over 30  $RSSR$  values are shown in the table.

### 5.4.3 Factor recovery in presence of noise

To show that our algorithm correctly recovers the factors, we adopted a visual illustration. Three  $64 \times 64$  images were treated as the loading matrices, and the original tensor was formed from these loading matrices according to Eq. (94). For example, the first image corresponds to loading matrix  $A$ , the second image to  $B$ , and the third image to  $C$ . An additional Gaussian noise with variance 1 was also added to the tensor. Afterwards, we ran each of the decomposition algorithms on the constructed tensor to recover the loading matrices. It should be noted that, this is an experiment for visual illustration of the factor

recovery using NTF decomposition and is not related to Tensorfaces or similar face recognition experiment. Figure 23 shows the original and the recovered loading matrices by various algorithms when they were executed for the same amount of time. It can be seen that by the time the **BPP** algorithm has recovered the factors successfully, other algorithms were yet to compute the factors.

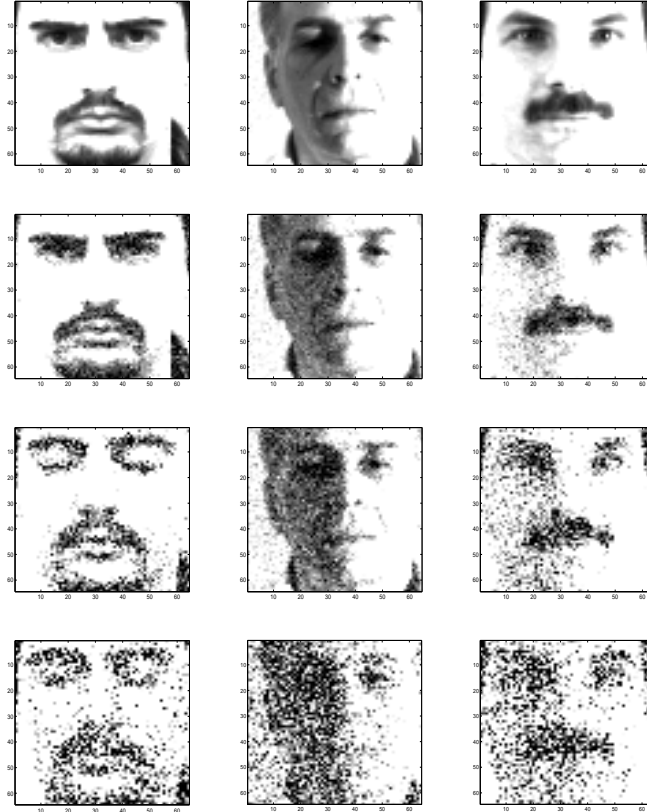
Numerical measures of the recovery in the presence of noise are also reported in Table 4. Timings shown in the table were obtained as follows. As **BPP** shows fastest trend in reducing the  $RSSR$  value as demonstrated in Figure 24, we first ran **BPP** until  $(RSSR(t-1) - RSSR(t))$  becomes smaller than  $10^{-6}$ . Then, all other algorithms were executed for the same amount of time, and their  $RSSR$  values are shown in the table. The results imply that under limited amount of computation time, **BPP** shows better recovery in general.

#### 5.4.4 Topic identification using NTF

In this subsection, we describe results of applying our NTF algorithm for topic identification.

*Group discussions in Enron data set:* We applied sparse NTF on the 3-way tensor derived from the Enron data set to identify topics and their participants. Sparsity constraints were enforced on two (sender and receiver) out of the three modes. Incorporating sparsity helps us to remove noise in the groups of users who discuss about a particular topic. For example, some users who were only once involved in the discussion on a particular topic should not be counted as a participant of the topic. Using sparse NTF helps in making the particular component in the factor zero. Sparse NTF with  $r = 10$  was applied, and the keywords of the topics and percentage of the users participating in each topic are summarized in Table 5.

*Scenario Discovery in IEEE VAST data set:* We used our NTF algorithm for scenario discovery in the IEEE Vast 2007 Contest data set to illustrate how term-entity-month based



**Figure 23:** Original images (1<sup>st</sup> row) and recovered images using **BPP** (2<sup>nd</sup> row), **AB** (3<sup>rd</sup> row), and **MU** (4<sup>th</sup> row) algorithms when they were computed for the same amount of time.

	$r$	<b>BPP</b>	ACTSET	<b>AB</b>	<b>MU</b>
Time(sec)	3	<b>0.955</b>	1.827	2.654	12.551
$RSSR$	3	<b>0.098</b>	0.098	0.099	0.142

**Table 2:** Timing comparison on amino acid data set:  $\mathbf{X} \in \mathbb{R}_+^{5 \times 201 \times 61}$ . The  $RSSR$  values were achieved by several algorithms at the corresponding amount of time.

NTF model can be used for scenario discovery. The data set consist of news stories and blog entries related to wildlife law enforcement, but mixed with some noisy information. Participants are asked to discover a major law enforcement/counter-terrorism scenario, form their hypotheses, and collect supporting evidences. In order to perform the task, the raw data was processed into a tensor format as described in [2]. We used a rank 25 approximation of the original tensor and listed the keywords of the discovered scenario in Table 6. We note that the detected topic is consistent with the ground truth revealed after the end of the contest. More details about this experiment can be found in [2].

$r$	Regularized NTF		Sparse NTF	
	<b>BPP</b>	<b>ACTSET</b>	<b>BPP</b>	<b>ACTSET</b>
10	<b>1.568</b>	3.144	<b>1.486</b>	2.921
50	<b>11.112</b>	23.091	<b>10.055</b>	21.991
100	<b>59.032</b>	92.543	<b>58.185</b>	90.321

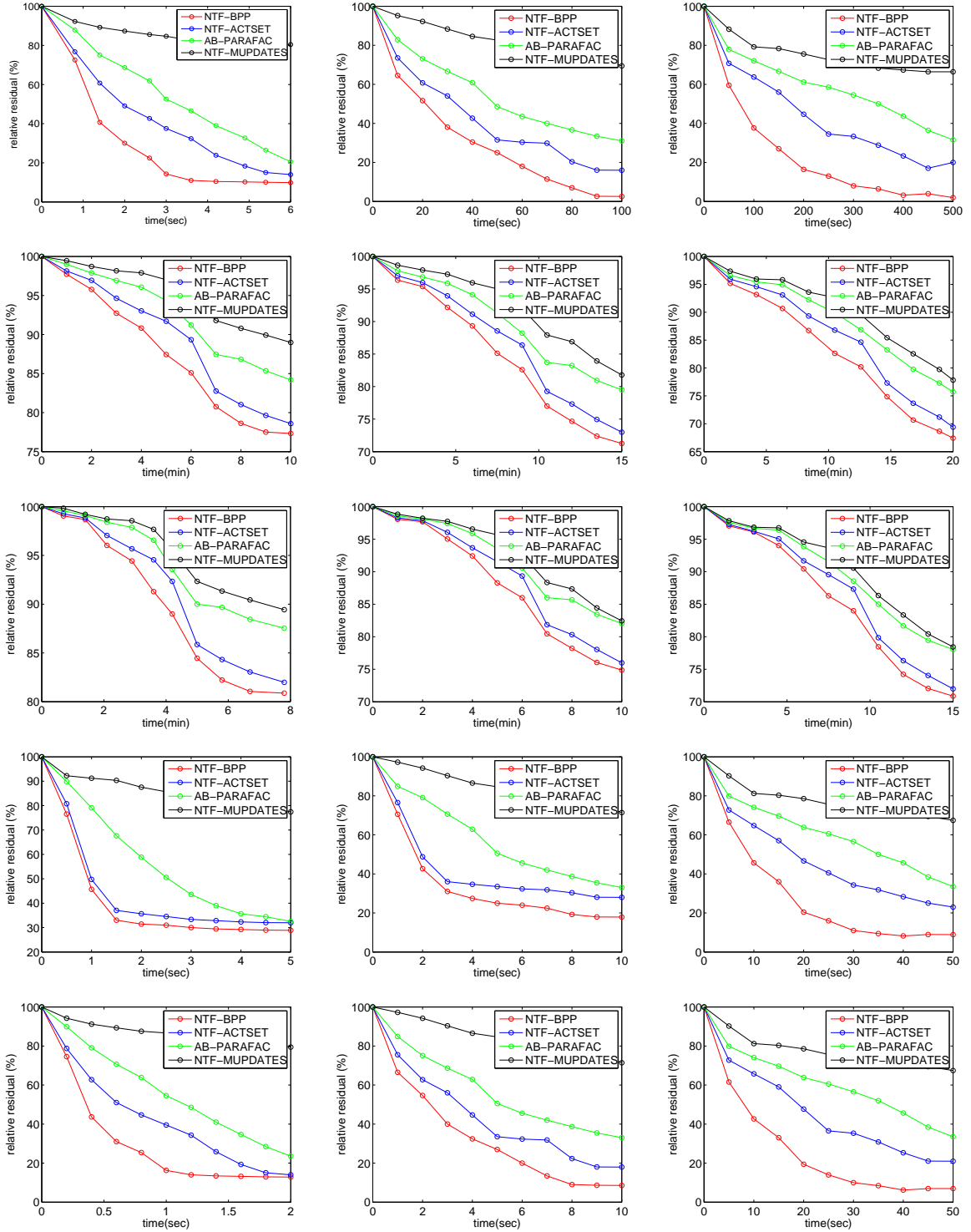
**Table 3:** Timing comparison on a synthetic tensor  $\mathbf{X} \in \mathbb{R}_+^{173 \times 234 \times 854}$ . Parameters used:  $\alpha = 0.4, \beta = 0.2$ , and  $\gamma = 0.06$  for regularized NTF, and  $\alpha = 0.5, \beta = 0.04$ , and  $\gamma = 0.2$  for sparse NTF.

$r$	$\sigma^2$	<b>BPP</b>	<b>ACTSET</b>	<b>AB</b>	<b>MU</b>
30	1.5	<b>0.239</b>	0.244	0.254	0.425
60	1.5	<b>0.063</b>	0.063	0.064	0.085
100	1.5	<b>0.009</b>	0.009	0.011	0.021
30	10	0.515	0.518	<b>0.513</b>	0.687
60	10	<b>0.438</b>	0.439	0.442	0.493
100	10	<b>0.173</b>	0.174	0.190	0.390

**Table 4:** Factor recovery under noise on two synthetic tensors in  $\mathbb{R}_+^{100 \times 221 \times 631}$  with Gaussian noise with two different variances. A smaller the  $RSSR$  value means better the recovery.

<b>Topics</b>	<b>% of users involved</b>
California legislature	4.3
India Dabhol Power Company	2.2
Downfall	16.3
Downfall(newsfeed)	9.3
El Paso Energy	5.3
Cal energy	7.8
TruOrange/College football	4.2
Enrononline	4.3
911 Sept 2001	5.6
NFL	2.3

**Table 5:** List of 10 topics identified by sparse NTF along with the percentage of people involved in the communication on each topic.



**Figure 24:** Timing-vs-RSSR graphs. The rows represent data sets: Enron 3-way (1<sup>st</sup> row), Enron 4-way (2<sup>nd</sup> row), VAST (3<sup>rd</sup> row), CMU facial (4<sup>th</sup> row), and synthetic (5<sup>th</sup> row) tensors. The columns represent reduced ranks:  $r = 10$  (left),  $r = 50$  (middle) and  $r = 90$  (right). The initialization of loading matrices were done randomly. The values reported in the graphs represent average value over 1,000 initializations for all data sets. For the synthetic data set (5<sup>th</sup>), the averaging was done over 1,000 synthetic tensor instances and 1,000 random initializations of loading matrices.

<b>Topics</b>
People for the Ethical Treatment of Animals, Animal rights, attacks on pet store
Monkeypox, Chinchilla
Arsons, Fire investigation
Drug trafficking, Exotic animal, Illegal trade
Animal Justice League, Protest activities
Animal treatment standards, Meat alternatives, Benefits for consumer health
Conservation of tigers, leopards
Wild chinchilla, Harvesting in Chile
Bullfighting, Cockfighting

**Table 6:** List of 9 topics identified by NTF from the VAST 2007 contest data set.

## CHAPTER VI

### DISCUSSION

#### *6.1 Learning without Labels*

We have demonstrated a collaborative framework for the estimation of classification and regression error rates for  $k \geq 1$  predictors. In contrast to previous supervised risk estimation methods such as cross validation [18], bootstrap [19], and others [22], proposed approach is fully unsupervised and thus able to use vast collections of unlabeled data. Other related work includes [50] and [46] which consider repeated labeling where each instance is labeled by multiple experts and the final label is decided based on a majority voting scheme. However, [50] and [46] fail to address estimating the risks of the predictors.

Also a novel framework for estimating margin-based risks using only unlabeled data was developed. We derived a theoretical basis by casting it as a maximum likelihood problem for Gaussian mixture model followed by plug-in estimation. Remarkably, the theory states that assuming normality of  $f_\theta(X)$  and a known  $p(Y)$  we are able to estimate the risk  $R(\theta)$  without a single labeled example. That is the risk estimate converges to the true risk as the number of unlabeled data increase. Moreover, using uniform convergence arguments it is possible to show that the proposed training algorithm converges to the optimal classifier as  $n \rightarrow \infty$  without any labeled data.

On a more philosophical level, our approach points at novel questions that go beyond supervised and semi-supervised learning. What benefit do labels provide over unsupervised training? Can it be extended to the multi-class case and to non-classification scenarios such as margin based regression or margin based structured prediction? When are the assumptions likely to hold and how can we make our framework even more resistant to deviations from them? These questions and others form new and exciting open research

directions.

## ***6.2 Nonnegative Tensor Factorization***

We presented a fast algorithm for nonnegative tensor factorization by decomposing the original minimization problem as a sequence of the NNLS problems. We use the block principal pivoting algorithm to efficiently solve the NNLS problems. Our algorithm provides a faster way of computing nonnegative tensor factorization and its regularized and sparse extensions. Experimental results show that the new algorithm is much faster than existing ones. We also demonstrate the applicability of our fast algorithm in analyzing large-scale multi-dimensional text data sets. Extending NTF for the case of missing entries, which can be used for semantics-based collaborative filtering is an interesting future direction.

## REFERENCES

- [1] ANDERSSON, C. and BRO, R., “The N-way toolbox for MATLAB,” *Chemometrics and Intelligent Laboratory Systems*, vol. 52, no. 1, pp. 1–4, 2000.
- [2] BADER, B., PURETSKIY, A., and BERRY, M., “Scenario discovery using nonnegative tensor factorization,” in *Progress in Pattern Recognition, Image Analysis and Applications, Proceedings of the Thirteenth Iberoamerican Congress on Pattern Recognition, CIARP*, pp. 791–805, Springer, 2008.
- [3] BALCAN, M. F. and BLUM, A., “A discriminative model for semi-supervised learning,” *Journal of the Association for Computing Machinery*, (to appear).
- [4] BEN-DAVID, S., LU, T., and PAL, D., “Does unlabeled data provably help? worst-case analysis of the sample complexity of semi-supervised learning,” in *International Conference on Learning Theory*, 2008.
- [5] BENTHEM, M. H. V. and KEENAN, M. R., “Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems,” *Journal of Chemometrics*, vol. 18, pp. 441–450, 2004.
- [6] BERK, K. N., “A central limit theorem for  $m$ -dependent random variables with unbounded  $m$ ,” *The Annals of Probability*, vol. 1, no. 2, pp. 352–354, 1973.
- [7] BERTSEKAS, D., HOMER, M., LOGAN, D., and PATEK, S., “Nonlinear programming,” *Athena scientific*, 1995.
- [8] BISHOP, Y., FIENBERG, S., and HOLLAND, P., *Discrete multivariate analysis: theory and practice*. MIT press, 1975.
- [9] BLITZER, J., DREDZE, M., and PEREIRA, F., “Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification,” in *Proc. of ACL '07*, 2007.
- [10] BREIMAN, L., “Bias, variance, and arcing classifiers,” Tech. Rep. 460, Statistics department, University of California, 1996.
- [11] BRO, R. and DE JONG, S., “A fast non-negativity-constrained least squares algorithm,” *Journal of Chemometrics*, vol. 11, no. 5, 1997.
- [12] CASTELLI, V. and COVER, T. M., “The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter,” *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 2102–2117, 1996.
- [13] CHAPELLE, O., SCHÖLKOPF, B., and ZIEN, A., eds., *Semi-Supervised Learning*. MIT Press, 2006.

- [14] COVER, T. M. and THOMAS, J. A., *Elements of Information Theory*. John Wiley & Sons, second ed., 2005.
- [15] COX, D., LITTLE, J., and O'SHEA, D., *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer, 2006.
- [16] DAI, W., YANG, Q., XUE, G.-R., and YU, Y., "Boosting for transfer learning," in *Proc. of International Conference on Machine Learning*, 2007.
- [17] DILLON, J. and LEBANON, G., "Statistical and computational tradeoffs in stochastic composite likelihood," in *Proc. of the 12th International Conference on Artificial Intelligence and Statistics*, 2009.
- [18] DUDA, R. O., HART, P. E., and STORK, D. G., *Pattern classification*. Wiley New York, 2001.
- [19] EFRON, B. and TIBSHIRANI, R. J., *An Introduction to the Bootstrap*. Chapman & Hall, 1997.
- [20] FERGUSON, T. S., *A Course in Large Sample Theory*. Chapman & Hall, 1996.
- [21] FRIEDLANDER, M. P. and HATZ, K., "Computing non-negative tensor factorizations," *Optimization Methods and Software*, vol. 23, no. 4, p. 631, 2008.
- [22] HAND, D. J., "Recent advances in error rate estimation," *Pattern Recognition Letters*, vol. 4, no. 5, pp. 335–346, 1986.
- [23] HARSHMAN, R., "Foundations of the PARAFAC procedure: Models and conditions for an explanatory multi-modal factor analysis," *UCLA working papers in phonetics*, vol. 16, pp. 1–84, 1970.
- [24] HÅSTAD, J., "Tensor rank is np-complete," *J. Algorithms*, vol. 11, no. 4, pp. 644–654, 1990.
- [25] HOEFFDING, W. and ROBBINS, H., "The central limit theorem for dependent random variables," *Duke Mathematical Journal*, vol. 15, pp. 773–780, 1948.
- [26] JOACHIMS, T., "Making large-scale svm learning practical," in *Advances in Kernel Methods - Support Vector Learning* (SCHÖLKOPF, B., BURGESS, C., and SMOLA, A., eds.), MIT Press, 1999.
- [27] KIM, H. and PARK, H., "Non-negative matrix factorization based on alternating non-negativity constrained least squares and active set method," *SIAM Journal in Matrix Analysis and Applications*, vol. 30, no. 2, pp. 713–730, 2008.
- [28] KIM, H., PARK, H., and ELDEN, L., "Non-negative Tensor Factorization Based on Alternating Large-scale Non-negativity-constrained," in *Proceedings of IEEE 7th International Conference on Bioinformatics and Bioengineering (BIBE07)*, pp. 1147–1151, 2007.

- [29] KIM, J. and PARK, H., “Toward faster nonnegative matrix factorization: A new algorithm and comparisons,” in *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM)*, pp. 353–362, 2008.
- [30] KOLDA, T. G. and BADER, B. W., “The TOPHITS model for higher-order web link analysis,” in *Proceedings of the Workshop on Link Analysis, Counterterrorism and Security*, 2006.
- [31] KOLDA, T. G. and BADER, B. W., “Tensor decompositions and applications,” *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.
- [32] LANG, K., “Newsweeder: Learning to filter netnews,” in *International Conference on Machine Learning*, 1995.
- [33] LATHAUWER, L. D., MOOR, B. D., and VANDEWALLE, J., “A multilinear singular value decomposition,” *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.
- [34] LEE, D. and SEUNG, H., “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, pp. 788–791, 1999.
- [35] LEE, D. and SEUNG, H., “Algorithms for non-negative matrix factorization,” in *Advances in Neural Information Processing Systems*, pp. 556–562, 2001.
- [36] LEWIS, D., YANG, Y., ROSE, T., and LI, F., “RCV1: A new benchmark collection for text categorization research,” *Journal of Machine Learning Research*, vol. 5, pp. 361–397, 2004.
- [37] LIANG, P. and JORDAN, M. I., “An asymptotic analysis of generative, discriminative, and pseudolikelihood estimators,” in *Proc. of the International Conference on Machine Learning*, 2008.
- [38] LIM, L. and COMON, P., “Nonnegative approximations of nonnegative tensors,” *Journal of Chemometrics*, vol. 23, pp. 432–441, 2009.
- [39] LIN, C., “Projected gradient methods for nonnegative matrix factorization,” *Neural Computation*, vol. 19, no. 10, pp. 2756–2779, 2007.
- [40] NIGAM, K., MCCALLUM, A., THRUN, S., and MITCHELL, T., “Text classification from labeled and unlabeled documents using EM,” *Machine Learning*, vol. 39, no. 2, pp. 103–134, 2000.
- [41] PAPOULIS, A., *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1984.
- [42] PHAM, T. V., WORRING, M., and SMEULDERS, A. W. M., “Face detection by aggregated bayesian network classifiers,” *Pattern Recognition Letters*, vol. 23, pp. 451–461, February 2002.

- [43] PORTUGAL, L., JUDICE, J., and VICENTE, L., “A comparison of block pivoting and interior-point algorithms for linear least squares problems with nonnegative variables,” *Mathematics of Computation*, pp. 625–643, 1994.
- [44] SHASHUA, A. and HAZAN, T., “Non-negative tensor factorization with applications to statistics and computer vision,” in *Proceedings of the 22nd international conference on Machine learning (ICML)*, 2005.
- [45] SHASHUA, A. and LEVIN, A., “Linear image coding for regression and classification using the tensor-rank principle,” in *Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.
- [46] SHENG, V. S., PROVOST, F., and IPEIROTIS, P. G., “Get another label? improving data quality and data mining using multiple, noisy labelers,” in *Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 614–622, 2008.
- [47] SILVA, V. D. and LIM, L.-H., “Tensor rank and the ill-posedness of the best low-rank approximation problem,” *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1084–1127, 2008.
- [48] SINGH, A., NOWAK, R., and ZHU, X., “Unlabeled data: Now it helps, now it doesn’t,” in *Advances in Neural Information Processing Systems*, vol. 22, 2008.
- [49] SINHA, K. and BELKIN, M., “The value of labeled and unlabeled examples when the model is imperfect,” in *Advances in Neural Information Processing Systems 20*, 2008.
- [50] SMYTH, P., FAYYAD, U., BURL, M., PERONA, P., and BALDI, P., “Inferring ground truth from subjective labelling of venus images,” in *Advances in Neural Information Processing Systems 7*, 1995.
- [51] SNOW, R., O’CONNOR, B., JURAFSKY, D., and NG, A. Y., “Cheap and fast-but is it good? evaluating non-expert annotations for natural language tasks,” in *Proc. of EMNLP*, 2008.
- [52] STURMFELS, B., *Solving Systems of Polynomial Equations*. American Mathematical Society, 2002.
- [53] SUN, J., PAPADIMITRIOU, S., LIN, C., CAO, N., LIU, S., and QIAN, W., “Multivis: Content-based social network exploration through multi-way visual analysis,” in *Proceedings of the 2009 SIAM International Conference on Data Mining (SDM)*, vol. 9, pp. 1063–1074, 2009.
- [54] TEICHER, H., “Identifiability of finite mixtures,” *The Annals of Mathematical Statistics*, vol. 34, no. 4, pp. 1265–1269, 1963.
- [55] TIBSHIRANI, R., “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

- [56] VAPNIK, V. N., *The Nature of Statistical Learning Theory*. Springer, second ed., 2000.
- [57] VASILESCU, M. and TERZOPOULOS, D., “Multilinear analysis of image ensembles: Tensorfaces,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 447–460, 2002.
- [58] WELLING, M. and WEBER, M., “Positive tensor factorization,” *Pattern Recognition Letters*, vol. 22, no. 12, pp. 1255–1261, 2001.