LEARNING MATRIX AND FUNCTIONAL MODELS IN HIGH-DIMENSIONS

A Thesis
Presented to
The Academic Faculty

by

Krishnakumar Balasubramanian

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Computer Science

Georgia Institute of Technology
August 2014

LEARNING MATRIX AND FUNCTIONAL MODELS IN HIGH-DIMENSIONS

Approved by:

Professor Guy Lebanon, Advisor
School of Computational Science and Engineering
Georgia Institute of Technology

Professor Maria-Florina Balcan
School of Computer Science
Georgia Institute of Technology

Professor John Lafferty
Department of Statistics and Computer Science
University of Chicago

Professor Le Song
School of Computational Science and Engineering
Georgia Institute of Technology

Professor Ming Yuan
Department of Statistics
University of Wisconsin

Date Approved: June 16, 2014

Dedicated to my parents, brother, bbkks and mlga.

ACKNOWLEDGEMENTS

I would like to thank my advisor Guy Lebanon for his guidance through my time in graduate school and providing me intellectual freedom to explore my research interests as it evolved. His inputs during the initial years helped me get started with research. I would like to thank all my committee members for their valuable comments regarding the thesis. Ming Yuan has helped me in more ways than I could have possibly imagined, both with research and other technical aspects. Discussions with John Lafferty have always provided me new perspectives and I wish to collaborate with him in future. Vladimir Koltchinskii has been a source of inspiration, through his research and excellent teaching. Though not related to this thesis, discussions with him have been extremely useful and without a doubt inspired a lot of my current (and future) research. I would also like to thank Kai Yu, for an enjoyable and useful internship opportunity and Bharath Sriperumbudur, who has been amazing to collaborate with. Special thanks to my friends and fellow students at lab 1305 for all the fun times and amazing discussions.

TABLE OF CONTENTS

DEDICATION . . . . . **iii**

ACKNOWLEDGEMENTS . . . . . **iv**

LIST OF TABLES . . . . . **x**

LIST OF FIGURES . . . . . **xii**

SUMMARY . . . . . **xviii**

I OVERVIEW OF THE THESIS **1**

1.1 Introduction 1

1.1.1 The Machine learning pipeline 1

1.2 Thesis Statement 3

1.3 Estimation and Prediction Problems in Non-standard Situations . . 4

1.3.1 Supervised Learning with No Labels 5

1.3.2 Optimal Random Effects Model for Multi-task Learning . . . 5

1.3.3 Landmark Selection Method for Multiple Output Prediction . 6

1.4 Learning Meaningful Data Representation and Feature Designing . . 6

1.4.1 Smooth Sparse Coding 7

1.4.2 RKHS Embedding based high Dimensional Feature Screening 8

II MARGIN-BASED CLASSIFICATION WITHOUT LABELS . . **9**

2.1 Introduction 9

2.2 Unsupervised Risk Estimation 11

2.2.1 Asymptotic Normality of $f_\theta(X)|Y$ 14

2.2.2 Unsupervised Consistency of $\hat{R}_n(\theta)$ 20

2.2.3 Unsupervised Consistency of $\arg\min \hat{R}_n(\theta)$ 22

2.2.4 Asymptotic Variance 25

2.2.5 Multiclass Classification 27

2.3 Application 1: Estimating Risk in Transfer Learning 29

2.4 Application 2: Unsupervised Learning of Classifiers 31

2.4.1 Inaccurate Specification of $\mathbb{P}(Y)$ . . . 37
2.4.2 Effect of Regularization and Dimensionality reduction. 38
2.5 Related Work . . . 39
2.6 Computing $M_{m,n}$ for Section 2.2.4 . . . 41
III ESTIMATING CLASSIFICATION AND REGRESSION ERRORS WITHOUT LABELS . . . 44
3.1 Introduction . . . 44
3.2 Unsupervised Risk Estimation Framework . . . 47
3.2.1 Non-Collaborative Estimation of the Risks . . . 49
3.2.2 Collaborative Estimation of the Risks: Conditionally Independent Predictors . . . 53
3.2.3 Collaborative Estimation of the Risks: Conditionally Correlated Predictors . . . 55
3.3 Extensions: Missing Values, Active Learning, and Semi-Supervised Learning . . . 57
3.4 Consistency of $\hat{\theta}_n^{\text{mle}}$ and $\hat{R}(f_j)$ . . . 58
3.4.1 Consistency of Classification Risk Estimation . . . 61
3.4.2 Consistency of Regression Risk Estimation . . . 68
3.5 Asymptotic Variance of $\hat{\theta}_n^{\text{mle}}$ and $\hat{R}$ . . . 70
3.6 Optimization Algorithms . . . 72
3.7 Empirical Evaluation . . . 73
3.8 Discussion . . . 84
IV LANDMARK SELECTION METHOD FOR MULTIPLE OUTPUT PREDICTION . . . 85
4.1 Introduction . . . 85
4.2 Related Work . . . 87
4.3 The landmark selection method: . . . 88
4.3.1 Step 1: Selecting the landmark set L and modeling (73) . . . 88
4.3.2 Step 2: Estimating (72) . . . 90
4.3.3 Step 3: Prediction . . . 91

4.4 Theory . . . 91
4.5 Optimization procedure . . . 94
4.6 Experiments . . . 95
4.6.1 Synthetic experiments . . . 95
4.6.2 Real-world data sets . . . 98

V OPTIMAL RANDOM EFFECTS MODEL FOR SPARSE MULTI-TASK LEARNING . . . 102

5.1 Introduction . . . 102
5.2 Joint Sparsity Random Effects Model and Group Lasso . . . 105
5.3 Joint Sparsity via Covariance Estimation . . . 106
5.3.1 Sparse Covariance Coding Models . . . 107
5.4 Theoretical Analysis . . . 108
5.4.1 Prediction Error . . . 109
5.4.2 Drawback of Group Lasso . . . 113
5.4.3 Other Covariance Coding Models . . . 116
5.5 Identifiability of additive structure . . . 117
5.6 Experiments . . . 118
5.6.1 Multi-task learning . . . 118
5.6.2 SCC based Image Classification . . . 122
5.6.3 Landmark selection . . . 123
5.7 A joint framework for covariance and regression co-efficient estimation 123

VI SMOOTH SPARSE CODING . . . 126

6.1 Introduction . . . 126
6.2 Related work . . . 127
6.3 Smooth Sparse Coding . . . 128
6.3.1 Spatio-Temporal smoothing . . . 130
6.4 Marginal Regression for Smooth Sparse Coding . . . 130
6.5 Sample Complexity of Smooth sparse coding . . . 134
6.6 Experiments . . . 137

6.6.1 Speed comparison . . . 138
6.6.2 Experiments with Kernel in Feature space . . . 138
6.7 Semi-supervised smooth sparse coding . . . 142
6.8 Data set Description . . . 143
6.8.1 CMU Multi-pie face recognition: . . . 143
6.8.2 15 Scenes Categorization: . . . 143
6.8.3 Caltech-101 Data set: . . . 144
6.8.4 Activity recognition . . . 144
6.8.5 Youtube person data set . . . 145
6.9 Experiments using Temporal Smoothing . . . 145
6.10 Generalization bounds for learning problems . . . 146
VII FEATURE SCREENING VIA RKHS EMBEDDINGS . . . 148
7.1 Introduction . . . 148
7.2 RKHS embedding of probabilities . . . 150
7.3 Screening via RKHS embedding . . . 152
7.3.1 DC-SIS as a special case of *sup*-HSIC-SIS . . . 154
7.4 Theoretical analysis . . . 155
7.4.1 Upper bounding the cardinality of $\widehat{\mathcal{M}}$ . . . 161
7.5 Iterative Screening procedures . . . 162
7.5.1 Method 1 . . . 162
7.5.2 Method 2 . . . 163
7.6 Experiments . . . 164
7.6.1 Synthetic data – univariate response . . . 164
7.6.2 Synthetic data – multivariate response . . . 165
7.6.3 Synthetic data – Iterative screening . . . 166
7.6.4 Gene array data set . . . 168
7.6.5 Multi-label classification data set . . . 168

VIII CONCLUSION **170**

8.1 Summary and Key Contributions 170

8.2 Related open problems 172

8.2.1 Joint Regularization for Multiple Low-rank Estimation 172

8.2.2 Sparse-additive Near-separable Nonnegative Matrix Factorization 173

REFERENCES **174**

VITA **184**

LIST OF TABLES

1 Comparison (test set error rate) between supervised logistic regression, Unsupervised logistic regression and Gaussian mixture modeling in original data space. The unsupervised classifier performs better than the GMM clustering on the original space and compares well with its supervised counterpart on most data sets. See text for more details. The stars represent GMM with covariance $\sigma^2 I$ due to the high dimensionality. In all other cases we used a diagonal covariance matrix. Non-diagonal covariance matrix was impractical due to the high dimensionality. 37

2 Test set Hamming loss and F1 measure evaluation of the four classification approaches: mlcs, ml-cca, one vs. all, and moplms. The base classifiers in the reduced space were SVM. 99

3 Test prediction error (MSE) for moplms vs. Lrmvr. λ_1 and λ_2 in state 1 were selected to minimize prediction error using cross-validation. The number of subproblems selected in this case was 98. 101

4 Support selection: Hamming distance between true non-zero indices and estimated non-zero indices by the indicated method for all signals. 119

5 Coefficient estimation: Normalized L_2 distance between true coefficients and estimated coefficients by the indicated method. First 5 rows correspond to 80% shared basis and the last 5 rows correspond to fully shared basis. 120

6 Coefficient estimation: Normalized L_2 distance between true coefficients and estimated coefficients by the indicated method with correlated input data. 120

7 Multi-task learning: Average (across task) MSE error on the test data set. 121

8 Face image classification based on gender: Test and Train set error rates for sparse covariance coding and group sparse coding (both with a fixed dictionary). 122

9 Simultaneous basis selection for data approximation: Average reconstruction error 124

10 Time comparison of coefficient learning in SC and SSC with either Lasso or Marginal regression updates. The dictionary update step was same for all methods. 139

11 Test set error accuracy for face recognition on CMU-multipie data set (left) 15 scene (middle) and Caltech-101 (right) respectively. The performance of the smooth sparse coding approach is better than the standard sparse coding and LLC in all cases. 140

12 Effect of dictionary size on classification accuracy using smooth sparse coding and marginal regression on 15 scene and Caltech -101 data set. 140

13 Action recognition (accuracy) for cited method (left), Hog3d+ SC (middle) and Hog3d+ SSC (right): KTH data set(top) YouTube action dataset (bottom). 141

14 Semi-supervised learning test set error: Dictionary learned from both CMU multi-pie and faces-on-tv data set using feature similarity kernel, used to construct sparse codes for CMU multipie data set. 143

15 Linear SVM accuracy for person recognition task from YouTube face video dataset. 145

16 Probability of support recovery using the distance kernel and Gaussian kernel: First four rows correspond to $P(\mathcal{M}^* = \widehat{\mathcal{M}})$ (corresponding to models 1, 2, 3 and 4 respectively) and the last four rows correspond to $P(\mathcal{M}^* \subset \widehat{\mathcal{M}})$. The very last row corresponds to the average cardinality of selected set. 166

17 Probability of support recovery using the distance kernel and Gaussian kernel. First two rows correspond to $P(\mathcal{M}^* = \widehat{\mathcal{M}})$ and the last three rows correspond to $P(\mathcal{M}^* \subset \widehat{\mathcal{M}})$. The very last row corresponds to the average cardinality of selected set over all experiments. 167

18 Advantage of iterative methods over sup-HSIC-SIS. The values reported are estimates of $P(\mathcal{M}^* \subset \widehat{\mathcal{M}})$ over 1000 trials. 167

19 Gene data set: Cardinality of selected set and predictive error (PE) under an additive model. 169

20 Test set classification error on the multi-label data sets. The number in the bracket correspond to the cardinality of selected feature set. . . 169

LIST OF FIGURES

1 Centered histograms of $f_\theta(X)|\{Y = 1\}$ overlayed with the pdf of a fitted Gaussian for randomly drawn θ vectors ($\theta_i \sim U(-1/2, 1/2)$). The columns represent datasets (RCV1 text data [67], MNIST digit images, and face images [80]) and the rows represent multiple random draws. For uniformity we subtracted the empirical mean and divided by the empirical standard deviation. The twelve 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 often for randomly drawn θ. 15

2 Centered histograms of $f_\theta(X)|\{Y = 1\}$ overlayed with the pdf of a fitted Gaussian for multiple θ vectors (four rows: 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 [67], MNIST digit images, and face images [80]). For uniformity we subtracted the empirical mean and divided by the empirical standard deviation. The twelve 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 for fitted θ values (except perhaps for L_1 regularization in the last row which promotes sparse θ). 16

3 Centered histograms of $f_\theta(X)|\{Y = 1\}$ overlayed with the pdf of a fitted Gaussian for multiple θ vectors (four rows: 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: USPS Handwritten Digits, Arcene data set, and ISOLET). For uniformity we subtracted the empirical mean and divided by the empirical standard deviation. The twelve panels further confirm that the assumption that $f_\theta(X)|Y$ is normal holds well for fitted θ values (except perhaps for L_1 regularization in the last row which promotes sparse θ) for various data sets. 17

4 Left panel: asymptotic accuracy (inverse of trace of asymptotic variance) of $\hat{R}_n(\theta)$ for logloss as a function of the imbalance of the class marginal $\mathbb{P}(Y)$. The accuracy increases with the class imbalance as it is easier to separate the two mixture components. Right panel: asymptotic accuracy (inverse of trace of asymptotic variance) as a function of the difference between the means $|\mu_1 - \mu_{-1}|$ and the variances σ_1/σ_2. See text for more information. 28

5 The relative accuracy of $\hat{R}_n$ (measured by $|\hat{R}_n(\theta) - R_n(\theta)|/R_n(\theta)$) as a function of n, classifier accuracy (acc) and the label marginal $\mathbb{P}(Y)$ (left: logloss, right: hinge-loss). The estimation error nicely decreases with n (approaching 1% at $n = 1000$ and decaying further). It also decreases with the accuracy of the classifier (top) and non-uniformity of $\mathbb{P}(Y)$ (bottom) in accordance with the theory of Section 2.2.4. . . . 30

6 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 by [30] which may be referred to for more detail. The train and testing sets contained samples from two top categories in the topic hierarchy but with different subcategory proportions. The first column indicates the top category classification task and the second indicates the empirical log-loss R_n calculated using the true labels of the testing set (5). The third and forth columns indicate the absolute and relative errors of $\hat{R}_n$. The fifth and sixth columns indicate the train set size and the label marginal distribution. 31

7 Performance of unsupervised logistic regression classifier $\hat{\theta}_n$ computed using Algorithm 1 (left) and Algorithm 2 (right) on the RCV1 dataset. The top two rows show the decay of the two risk estimates $\hat{R}_n(\hat{\theta}_n)$, $R_n(\hat{\theta}_n)$ as a function of the algorithm iterations. The risk estimates of $\hat{\theta}_n$ were computed using the train set (top) and the test set (middle). The bottom row displays the decay of the test set error rate of $\hat{\theta}_n$ as a function of the algorithm iterations. The figure shows that the algorithm obtains a relatively accurate classifier (testing set error rate 0.1, and $\hat{R}_n$ decaying similarly to R_n) without the use of a single labeled example. For comparison, the test error rate for supervised logistic regression with the same n is 0.07. 34

8 Performance of unsupervised logistic regression classifier $\hat{\theta}_n$ computed using Algorithm 1 (left) and Algorithm 2 (right) on the MNIST dataset. The top two rows show the decay of the two risk estimates $\hat{R}_n(\hat{\theta}_n)$, $R_n(\hat{\theta}_n)$ as a function of the algorithm iterations. The risk estimates of $\hat{\theta}_n$ were computed using the train set (top) and the test set (middle). The bottom row displays the decay of the test set error rate of $\hat{\theta}_n$ as a function of the algorithm iterations. The figure shows that the algorithm obtains a relatively accurate classifier (testing set error rate 0.1, and $\hat{R}_n$ decaying similarly to R_n) without the use of a single labeled example. For comparison, the test error rate for supervised logistic regression with the same n is 0.05. 35

9 Performance of unsupervised classifier training on RCV1 data (top class vs. classes 2-5) for misspecified $\mathbb{P}(Y)$. The performance of the estimated classifier (in terms of training set empirical logloss R_n (5) and test error rate measured using held-out labels) decreases with the deviation between the assumed and true $\mathbb{P}(Y = 1)$ (true $\mathbb{P}(Y = 1) = 0.3$)). The classifier performance is very good when the assumed $\mathbb{P}(Y)$ is close to the truth and degrades gracefully when the assumed $\mathbb{P}(Y)$ is not too far from the truth. 38

10 Test set error rate versus regularization parameter (L_2 on the left panel and L_1 on the right panel) for supervised and unsupervised logistic regression on RCV1 data set. 39

11 Test set error rate versus the amount of dimensions used (extracted via PCA) for supervised and unsupervised logistic regression on USPS data set. The original dimensionality was 256. 40

12 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 $\mathbb{P}(Y = 1) = 0.75$. 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 θ^{true}. 54

13 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 θ^{true} resulting in a more accurate risk estimate. 55

14 Left: Average value of $|\hat{\theta}_n^{\text{mle}} - \theta^{\text{true}}|$ as a function of θ^{true} and $\mathbb{P}(Y = 1)$ for $k = 1$ classifier and $n = 500$ (computed over a uniform spaced grid of 15×15 points). The plot illustrates the increased accuracy obtained by a less uniform $\mathbb{P}(Y)$. Right: Fisher information $J(\theta)$ for $k = 1$ as a function of θ^{true} and $\mathbb{P}(Y)$. The asymptotic variance of the estimator is $J^{-1}(\theta)$ which closely matches the experimental result in the left panel. 74

15 Left: Scatter plot contrasting the true and predicted values of θ in the case of a single classifier $k = 1$, $\mathbb{P}(Y = 1) = 0.8$, 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 θ^{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 ($\theta_i^{\text{true}} = \mathbb{P}(Y = 1) = 0.75$) in the collaborative case. The estimation error decreases as more classifiers are used due to the collaborative nature of the estimation process. 75

16 Left: Scatter plot contrasting the true and predicted values of θ 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 θ^{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. 76

17 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 $\mathbb{P}(Y = 1) = 0.75$. 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. 77

18 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. 78

19 The figure compares the estimator accuracy assuming that the marginal $\mathbb{P}(Y)$ 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) 78

20 Mean prediction accuracy for the unsupervised predictor combination scheme in (25) 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. 79

21 $\text{mae}(\hat{\theta}^{\text{mle}}, \theta^{\text{true}})$ as a function of n for different number of annotators k on RTE (left) and TEMP (right) datasets. Left: $n = 100$, $\mathbb{P}(Y = 1) = 0.5$ and $\theta^{\text{true}} = \{0.85, 0.92, 0.58, 0.5, 0.51\}$. Right: $n = 190$, $\mathbb{P}(Y = 1) = 0.56$ and $\theta^{\text{true}} = \{0.93, 0.92, 0.54, 0.44, 0.92\}$. The classifiers were added in the order specified. 80

22 $\text{mae}(\theta^{\text{true}}, \hat{\theta}^{\text{mle}})$ as a function of the test set size on the Ringnorm dataset. $\mathbb{P}(Y = 1) = 0.47$, and θ^{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. 82

23 $\text{mae}(\hat{\theta}^{\text{mle}}, \theta^{\text{true}})$ for the domain adaptation ($n = 1000$, $\mathbb{P}(Y = 1) = 0.75$) and 20 newsgroup ($n = 15,000$, $\mathbb{P}(Y = 1) = 0.05$ 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. 83

24 Left: MSE vs. sample size for synthetic regression data set. Middle: MSE vs. sample size for synthetic regression data set. Right: Hamming loss as a function of sample size for synthetic classification data set. The multiple curves represent different values of s/k. 96

25 Hamming loss vs. sample size on synthetic classification data sets. . . 96

26 Left and middle: Hamming loss versus number of samples for moplms, mlcs and ml-cca on delicious data set (left) and image data set (middle). Right: Mean MSE prediction error as a function of sample size for moplms, low rank multivariate regression and group Lasso based multivariate regression. 100

27 Comparison between the histograms of Fisher discriminant score realized by sparse coding and smooth sparse coding. The images represent the histogram of the ratio of smooth sparse coding Fisher score over standard sparse coding Fisher score (left: image data set; right: video). A value greater than 1 implies that smooth sparse coding is more discriminatory. 141

SUMMARY

A large amount of data has been generated over the last decade. The size and dimensionality of the data sets have grown at an unprecedented rate. As a result, the task of developing methods to analyze and extract information from such data sets has become crucial. A caveat is that, with the amount of data available one can essentially discover any pattern one wants to, irrespective of it being true or false. Hence the challenge is to develop efficient procedure to extract information that is *meaningful* from a scientific perspective.

Modern statistical methods provide us with a principled framework for extracting such *meaningful* information from noisy high-dimensional data sets. A significant feature of such procedures is to be able to make inferences from the data that are statistically significant and computationally efficient. In this thesis we make several contributions to such statistical procedures. Our contributions are two-fold.

We first address prediction and estimation problems. A particular drawback of existing approaches is that they are not designed to handle certain non-standard situations that arise in practice. Specifically, in order to evaluate or train a predictor, labeled data is required by existing methods. While labeled data is typically expensive to obtain, it might be relatively easy and inexpensive to obtain large amounts of unlabeled data. Also, in several situations labeled data may not be available at all, for example, due to privacy reasons. We develop principled procedures that enable one to train and evaluate predictors, provably well in those situations. We also address prediction with large output spaces and develop procedures that predict with complexity logarithmic in the dimensionality of the output space. Furthermore, we propose an asymptotically optimal procedure for sparse multi-task learning under a

random effects model.

We next address the problem of learning meaningful representation of data. The task of feature design for the subsequent estimation/prediction problem has been shown to be of at most importance to gain better performance. Towards that, we develop a new procedure for obtaining sparse representations of data, that takes into account the spatial structure of the data space. We also develop a method to obtain sparse representations for several related tasks with shared structure. Next, we develop a model-free method for selecting a relevant subset of features given a large number of features.

In summary, our contributions add to the existing set of statistical procedures for extracting *meaningful* information from large data sets. It also extends the applicability of such procedures to several previously unstudied scenarios and helps obtain better performance.

CHAPTER I

OVERVIEW OF THE THESIS

1.1 Introduction

The focus of this thesis is on developing and analyzing statistical machine learning algorithms for data analysis. Two major components of such information systems are data and models. Modern data sets have grown in size at a relatively fast rate. They are large-scale and often high-dimensional. The success of building information systems capable of deriving meaningful insights from such modern data sets, lies crucially on coming up with sophisticated models of learning. Under the well established minimax framework for statistical inference, one could note that simple models are fundamentally limited in their information extraction capabilities and existing non-parametric methods are inefficient in high-dimensions, as they suffer from the so called curse of dimensionality.

The opportunities provided by the data deluge and limitations exhibited by the simple models, naturally point at developing more sophisticated, yet efficient models of learning in high-dimensions. Since the developed models would have limited usage if they are computationally demanding, computational feasibility is a fundamental requirement. The thesis aims at developing such computationally and statistically efficient estimation procedures based on matrix and functional models for analyzing large-scale high-dimensional data sets.

1.1.1 The Machine learning pipeline

A standard and highly successful framework for applying statistical learning techniques for information extraction problems involves two steps: 1) estimation/prediction given the features and 2) designing features or learning representations from data.

More precisely, given input $X \in \mathcal{X}$ and output $Y \in \mathcal{Y}$ with a joint distribution $\mathbb{P}(X, Y)$, the task of prediction is to find a measurable mapping $f : \mathcal{X} \mapsto \mathcal{Y}$ that minimizes the risk given by $\mathsf{E}\,\mathcal{L}(f(X), Y)$, where $\mathcal{L}$ is a (prediction/estimation) loss function that penalizes the error made by the mapping f.

In practice, we do not observe the distribution $\mathbb{P}(X, Y)$, but we are given access to training samples $\{X^{(i)}, Y^{(i)}\}_{i=1}^{n}$, with $X^{(i)} \in \mathcal{X}$ and $Y^{(i)} \in \mathcal{Y}$ for all $i = 1, \ldots, n$. In this case, step (1) consists of estimating a mapping $\hat{f}$ that minimizes the empirical risk defined as $n^{-1} \sum_{i=1}^{n} \mathcal{L}(f(X^{(i)}), Y^{(i)})$. Early statistical research focused mainly on this prediction/estimation part of the pipeline. Furthermore, their scope was limited in the sense that, they operated under ideal situations (as will be described in detail in the subsequent sections). Modern problems and data sets pose a different set of challenges that often cannot be handled by those methods. Some examples are: no access to output samples (i.e., we are given access to only$\{X^{(i)}\}_{i=1}^{n}$); high-dimensionality of the data sets (i.e., $\mathcal{X} \subseteq \mathbb{R}^d$, with $d > n$) which necessitates developing variable selection techniques; large output-spaces (i.e., $\mathcal{Y} \subseteq \mathbb{R}^k$ with $k > n$) which bring about statistical and computational challenges for efficient prediction.

Furthermore feature designing or representation learning, has emerged as an important step that one should focus on, while designing statistical learning systems to have improved performance. A main motivation is that in many practical cases, getting access to large amounts of unlabeled data is easy compared to getting access to labeled data. This rises the question if one could use the unlabeled samples to learn some representations of the data, that might improve the subsequent prediction/estimation step. Representation learning, at a high-level, involves several techniques for extracting or learning efficient features from the given data. More precisely, given input sample $X \in \mathcal{X}$, the goal of representation learning is to find a mapping (from a finite dimensional space for simplicity) $T : \mathbb{R}^p \mapsto \mathcal{X}$ and a code $C \in \mathbb{R}^p$, such that the following reconstruction error (with respect to the loss ℓ_r is

minimum: $\ell_r(X, T(C))$. When $\mathcal{X} \subseteq \mathbb{R}^d$, depending on the structure of the mapping T and relationship between p and d, several representation learning techniques are defined.

Several theoretical and empirical justifications exist for adopting such feature design methods. With the learned representation of the data, one could proceed to do prediction and estimation tasks. These representations may or may not have specific interpretations. For example, given a face image, the learned representation of the image may not necessarily correspond to another face image. In some cases though, the learned representations need to have a particular interpretation. In this case, the feature design boils down to selecting a subset of features from the given features. A canonical example is the variable selection method (for example, Lasso) used in gene expression data sets.

1.2 Thesis Statement

In this thesis, we focus on both parts of the above mentioned pipeline. Specifically,

> *we develop principled procedures for estimating, predicting and learning efficient representations from high-dimensional data in several non-standard and challenging situations.*

The developed procedures enable one to design and analyze better statistical learning systems. Towards that, we make the following contributions:

- We rigorously show that even when given no access to labeled samples, one can still consistently estimate error rate of predictors and train predictors with respect to a given (convex) loss function. We derive consistency of the proposed method given high-dimensional data.

- We propose an efficient procedure for predicting with large output spaces that

scales logarithmically in the dimensionality of the output space. We demonstrate that the method outperforms existing methods in the considered regime.

- We device an optimal procedure for performing multi-task learning when the tasks share a joint support. We show the consistency of the proposed method and derive rates of convergence. Furthermore the same approach enables one to design features for multiple related tasks.

- We propose a method for learning sparse features that takes into account the structure of the data space and demonstrates how it enables one to obtain better features compared to existing methods. We further establish sample complexity results for the proposed approach.

- We propose a model-free variable selection procedure and establish its sure-screening property in the high dimensional regime. The method is flexible and can handle non-standard and multivariate output spaces directly.

1.3 Estimation and Prediction Problems in Non-standard Situations

The first focus-point of the thesis aims at the fundamental statistical task of estimation and prediction, given the data representation. While a large body of successful work exists on this problem, much of the methods developed assumes and works under ideal or standardized situations (for example, availability of complete data set). Compared to previous work, we consider statistical inference under several non-standard, yet practically frequently occurring situations where existing methods are no longer meaningful. Towards that, we first established a framework called *Unsupervised Supervised Learning*, which enables one to do supervised tasks (which normally require labels during training) like estimating error rates of predictors and training classifiers without labels. Next, we considered high-dimensional output spaces (for example,

consider an image tagging system where output space is of the order of 10^6) and developed an efficient landmark based functional prediction framework. This is yet another non-standard and challenging setting not often considered in existing work.

1.3.1 Supervised Learning with No Labels

Many popular linear classifiers, such as logistic regression, boosting, or SVM, are trained by optimizing a margin-based risk function. Traditionally, these risk functions are computed based on a labeled dataset. We develop a novel technique for estimating such risks using only unlabeled data and the marginal label distribution. We prove that the proposed risk estimator is consistent on high-dimensional datasets and demonstrate it on synthetic and real-world data. Furthermore, estimating the error rates of classifiers or regression models is a fundamental task in machine learning which has thus far been studied exclusively using supervised learning techniques. We propose a novel unsupervised framework for estimating these error rates using only unlabeled data and mild assumptions. We prove consistency results for the framework and demonstrate its practical applicability on both synthetic and real world data. This is joint work with Guy Lebanon and Pinar Donmez and is described in Chapters 2 and 3. The material of these chapters can also be found in the following published papers: [3] and [33].

1.3.2 Optimal Random Effects Model for Multi-task Learning

Joint sparsity regularization in multi-task learning has attracted much attention in recent years. The traditional convex formulation employs the group Lasso relaxation to achieve joint sparsity across tasks. Although this approach leads to a simple convex formulation, it suffers from several issues due to the looseness of the relaxation. To remedy this problem, we view jointly sparse multi-task learning as a specialized random effects model, and derive a convex relaxation approach that involves two steps. The first step learns the covariance matrix of the coefficients using a convex

formulation which we refer to as sparse covariance coding; the second step solves a ridge regression problem with a sparse quadratic regularizer based on the covariance matrix obtained in the first step. It is shown that this approach produces an asymptotically optimal quadratic regularizer in the multitask learning setting when the number of tasks approaches infinity. Experimental results demonstrate that the convex formulation obtained via the proposed model significantly outperforms group Lasso (and related multi-stage formulations). This is joint work with Kai Yu and Tong Zhang and is described in Chapter 5. The material of this chapter can also be found in the following published paper: [7].

1.3.3 Landmark Selection Method for Multiple Output Prediction

Conditional modeling $\mathcal{X} \mapsto \mathcal{Y}$ is a central problem in machine learning. A substantial research effort is devoted to such modeling when $\mathcal{X} \subset \mathbb{R}^d$ is high dimensional. We consider, instead, the case of a high dimensional $\mathcal{Y} \subset \mathbb{R}^k$, where $\mathcal{X}$ is either low dimensional or high dimensional. Our approach is based on selecting a small subset $\mathcal{Y}_L$ of the dimensions of $\mathcal{Y}$, and proceed by modeling (i) $\mathcal{X} \mapsto \mathcal{Y}_L$ and (ii) $\mathcal{Y}_L \mapsto \mathcal{Y}$. Composing these two models, we obtain a conditional model $\mathcal{X} \mapsto \mathcal{Y}$ that possesses convenient statistical properties. Multi-label classification and multivariate regression experiments on several datasets show that this method outperforms the one vs. all approach as well as several sophisticated multiple output prediction methods. This is joint work with Guy Lebanon and is described in Chapter 4. The material of this chapters can also be found in the following published paper: [4].

1.4 Learning Meaningful Data Representation and Feature Designing

While the previous line of work focused mainly on estimation and prediction problems with a given data representation, in this line of work we consider the problem of efficiently learning meaningful data representation in high-dimensions. We refer

the reader to [76] for the general feature representation framework, which includes standard methods like principal component analysis (PCA) and non-negative matrix factorization (NMF) and sparse coding (SC). Recent research has empirically shown the advantages of learning data representations for a variety of prediction tasks. Towards that, we proposed a smooth version of sparse coding that taking into account spatial and temporal information of the data, along with theoretical guarantees for the method. In some applications, one might just want to select relevant features from existing ones. Focusing on such applications, we developed a novel feature selection method, based on kernel embeddings, which does not assume any regressive model between the input and the output (and hence model-free).

1.4.1 Smooth Sparse Coding

We propose and analyze a novel framework for learning sparse representations, based on two statistical techniques: kernel smoothing and marginal regression. The proposed approach provides a flexible framework for incorporating feature similarity or temporal information present in data sets, via non-parametric kernel smoothing. We provide generalization bounds for dictionary learning using smooth sparse coding and show how the sample complexity depends on the L_1 norm of kernel function used. Furthermore, we propose using marginal regression for obtaining sparse codes, which significantly improves the speed and allows one to scale to large dictionary sizes easily. We demonstrate the advantages of the proposed approach, both in terms of accuracy and speed by extensive experimentation on several real data sets. In addition, we demonstrate how the proposed approach could be used for improving semi-supervised sparse coding. This is Joint work with Kai Yu and Guy Lebanon and is described in Chapter 6. The material of this chapter can also be found in the following published paper: [6].

1.4.2 RKHS Embedding based high Dimensional Feature Screening

Feature screening is a key step in handling ultrahigh dimensional data sets that are ubiquitous in modern statistical problems. Over the last decade, convex relaxation based approaches (e.g., Lasso/sparse additive model) have been extensively developed and analyzed for feature selection in high dimensional regime. But in the ultrahigh dimensional regime, these approaches suffer from several problems, both computationally and statistically. To overcome these issues, we propose a novel Hilbert space embedding based approach to independence screening for ultrahigh dimensional data sets. The proposed approach is model-free (i.e., no model assumption is made between response and predictors) and could handle non-standard (e.g., graphs) and multivariate outputs directly. We establish the sure screening property of the proposed approach in the ultrahigh dimensional regime, and experimentally demonstrate its advantages and superiority over other approaches on several synthetic and real data sets. This is joint work with Bharath Sriperumbudur and Guy Lebanon and is described in Chapter 7. The material of this chapter can also be found in the following published paper: [5].

CHAPTER II

MARGIN-BASED CLASSIFICATION WITHOUT LABELS

2.1 Introduction

In this chapter, we consider binary classification problem with the input space $\mathcal{X} \subset \mathbb{R}^d$ and the output space $\mathcal{Y} = \{-1, +1\}$. Many popular linear classifiers, such as logistic regression, boosting, or SVM, or their additive versions, are trained by optimizing a margin-based risk function. For standard linear classifiers $\hat{Y} = \text{sign} \sum \theta_j X_j$ with $Y \in \{-1, +1\}$, and $X, \theta \in \mathbb{R}^d$ the margin is defined as the product

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

Similarly, for additive classifiers, $\hat{Y} = \text{sign} \sum f_j(X_j)$ with $Y \in \{-1, +1\}$, and $X \in \mathbb{R}^d$ and $f_j : \mathbb{R} \mapsto \mathbb{R}$ are univariate functions assumed to be in a reproducing kernel Hilbert space or Sobolev space. The margin is defined similar to the linear case, as the product $Y f_\theta(X)$ where $f_\theta(X) \stackrel{\text{def}}{=} \sum_{j=1}^{d} f_j(X_j)$. We consider linear classifiers in this chapter, but the exposition applies to additive classifiers with minor modifications.

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

$$R(\theta) = \mathsf{E}_{\mathbb{P}(X,Y)} \mathcal{L}(Y, f_\theta(X)) \tag{1}$$

with the three most popular loss functions

$$\mathcal{L}_1(Y, f_\theta(X)) = \exp\left(-Y f_\theta(X)\right), \tag{2}$$

$$\mathcal{L}_2(Y, f_\theta(X)) = \log\left(1 + \exp\left(-Y f_\theta(X)\right)\right) \text{ and} \tag{3}$$

$$\mathcal{L}_3(Y, f_\theta(X)) = \left(1 - Y f_\theta(X)\right)_+ \tag{4}$$

being exponential loss $\mathcal{L}_1$ (boosting), logloss $\mathcal{L}_2$ (logistic regression) and hinge loss $\mathcal{L}_3$ (SVM) respectively (A_+ above corresponds to A if $A > 0$ and 0 otherwise).

Since the risk $R(\theta)$ depends on the unknown distribution $\mathbb{P}$, it is usually replaced during training with its empirical counterpart

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

based on a labeled training set

$$(X^{(1)}, Y^{(1)}), \ldots, (X^{(n)}, Y^{(n)}) \overset{\text{iid}}{\sim} \mathbb{P} \tag{6}$$

leading to the following estimator

$$\hat{\theta}_n = \arg\min_{\theta} R_n(\theta).$$

Note, however, that evaluating and minimizing R_n requires labeled data (6). While suitable in some cases, there are certainly situations in which labeled data is difficult or impossible to obtain.

In this chapter we construct an estimator for $R(\theta)$ using only unlabeled data, that is using

$$X^{(1)}, \ldots, X^{(n)} \overset{\text{iid}}{\sim} \mathbb{P} \tag{7}$$

instead of (6). Our estimator is based on the assumption that when the data is high dimensional ($d \to \infty$) the quantities

$$f_\theta(X) | \{Y = y\}, \quad y \in \{-1, +1\} \tag{8}$$

are normally distributed. 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 (8) may be estimated from unlabeled data (7) and that these distributions may be used to measure margin-based losses such as (2)-(4). 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 lead 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.

2.2 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 $R(\theta)$ defined in (1) using the unlabeled data (7) which we denote $\hat{R}_n(\theta\,; X^{(1)}, \ldots, X^{(n)})$ or simply $\hat{R}_n(\theta)$ (to distinguish it from R_n in (5)).

Our estimation is based on two assumptions. The first assumption is that the label marginals $\mathbb{P}(Y)$ are known and that $\mathbb{P}(Y = 1) \neq \mathbb{P}(Y = -1)$. While this assumption may seem restrictive at first, there are many cases where it holds. Examples include medical diagnosis ($\mathbb{P}(Y)$ is the well known marginal disease frequency), handwriting recognition or OCR ($\mathbb{P}(Y)$ is the easily computable marginal frequencies of different letters in the English language), life expectancy prediction ($\mathbb{P}(Y)$ is based on marginal life expectancy tables). In these and other examples $\mathbb{P}(Y)$ is known with great accuracy even if labeled data is unavailable. Our experiments show that assuming a wrong marginal $\mathbb{P}'(Y)$ causes a graceful performance degradation in $|\mathbb{P}(Y) - \mathbb{P}'(Y)|$. Furthermore, the assumption of a known $\mathbb{P}(Y)$ may be replaced with a weaker form in which we know the ordering of the marginal distributions e.g., $\mathbb{P}(Y = 1) > \mathbb{P}(Y = -1)$, but without knowing the specific values of the marginal distributions.

The second assumption is that the quantity $f_\theta(X)|Y$ follows a normal distribution. As $f_\theta(X)|Y$ is a linear combination of random variables, it is frequently normal when X is high dimensional. From a theoretical perspective this assumption is motivated by the central limit theorem (CLT). The classical CLT states that $f_\theta(X) = \sum_{i=1}^{d} \theta_i X_i|Y$ is approximately normal for large d if the data components $X_1, \ldots, X_d$ are iid given Y. A more general CLT states that $f_\theta(X)|Y$ is asymptotically normal if $X_1, \ldots, X_d|Y$ are independent (but not necessary identically distributed). Even more general CLTs state that $f_\theta(X)|Y$ is asymptotically normal if $X_1, \ldots, X_d|Y$ are not independent but their dependency is limited in some way. We examine this issue in Section 2.2.1 and also show that the normality assumption holds empirically for several standard datasets.

To derive the estimator we rewrite (1) by taking expectation with respect to Y and $\alpha = f_\theta(X)$

$$R(\theta) = \mathsf{E}_{\mathbb{P}(f_\theta(X),Y)} \mathcal{L}(Y, f_\theta(X)) = \sum_{y \in \{-1,+1\}} \mathbb{P}(y) \int_{\mathbb{R}} \mathbb{P}(f_\theta(X) = \alpha|y) \mathcal{L}(y, \alpha)\, d\alpha. \quad (9)$$

Equation (9) involves three terms $\mathcal{L}(y, \alpha)$, $\mathbb{P}(Y)$ and $\mathbb{P}(f_\theta(X) = \alpha|y)$. The loss function $\mathcal{L}$ is known and poses no difficulty. The second term $\mathbb{P}(Y)$ is assumed to be known (see discussion above). The third term is assumed to be normal $f_\theta(X) \,|\, \{Y = y\} = \sum_i \theta_i X_i \,|\, \{Y = y\} \sim N(\mu_y, \sigma_y)$ with parameters μ_y, σ_y, $y \in \{-1, 1\}$ that are estimated by maximizing the likelihood of a Gaussian mixture model (we denote $\mu = (\mu_1, \mu_{-1})$ and $\sigma^2 = (\sigma_1^2, \sigma_{-1}^2)$. These estimated parameters are used to construct the plug-in estimator $\hat{R}_n(\theta)$ as follows:

$$\ell_n(\mu, \sigma) = \sum_{i=1}^{n} \log \sum_{y^{(i)} \in \{-1,+1\}} \mathbb{P}(y^{(i)}) \mathbb{P}_{\mu_y, \sigma_y}(f_\theta(X^{(i)})|y^{(i)}). \quad (10)$$

$$(\hat{\mu}^{(n)}, \hat{\sigma}^{(n)}) = \arg\max_{\mu, \sigma} \ell_n(\mu, \sigma). \quad (11)$$

$$\hat{R}_n(\theta) = \sum_{y \in \{-1,+1\}} \mathbb{P}(y) \int_{\mathbb{R}} \mathbb{P}_{\hat{\mu}_y^{(n)}, \hat{\sigma}_y^{(n)}}(f_\theta(X) = \alpha|y) \mathcal{L}(y, \alpha)\, d\alpha. \quad (12)$$

We make the following observations.

1. Although we do not denote it explicitly, μ_y and σ_y are functions of θ.

2. The loglikelihood (45) does not use labeled data (it marginalizes over the label $y^{(i)}$).

3. The parameters of the loglikelihood (45) are $\mu = (\mu_1, \mu_{-1})$ and $\sigma = (\sigma_1, \sigma_{-1})$ rather than the parameter θ associated with the margin-based classifier. We consider the latter one as a fixed constant at this point.

4. The estimation problem (11) is equivalent to the problem of maximum likelihood for means and variances of a Gaussian mixture model where the label marginals are assumed to be known. It is well known that in this case (barring the symmetric case of a uniform $\mathbb{P}(Y)$) the MLE converges to the true parameter values [106].

5. The estimator $\hat{R}_n$ (12) is consistent in the limit of infinite unlabeled data

$$P\left(\lim_{n\to\infty} \hat{R}_n(\theta) = R(\theta)\right) = 1.$$

6. The two risk estimators $\hat{R}_n(\theta)$ (12) and $R_n(\theta)$ (5) approximate the expected loss $R(\theta)$. The latter uses labeled samples and is typically more accurate than the former for a fixed n.

7. Under suitable conditions $\arg\min_\theta \hat{R}_n(\theta)$ converges to the expected risk minimizer

$$P\left(\lim_{n\to\infty} \operatorname*{arg\,min}_{\theta\in\Theta} \hat{R}_n(\theta) = \operatorname*{arg\,min}_{\theta\in\Theta} R(\theta)\right) = 1.$$

This far reaching conclusion implies that in cases where $\arg\min_\theta R(\theta)$ is the Bayes classifier (as is the case with exponential loss, log loss, and hinge loss) we can retrieve the optimal classifier without a single labeled data point.

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

The quantity $f_\theta(X)|Y$ is essentially a sum of d random variables which under some conditions for large d is likely to be normally distributed. One way to verify this is empirically, as we show in Figures 1-3 which contrast the histogram with a fitted normal pdf for text, digit images, and face images data. For these datasets the dimensionality d is sufficiently high to provide a nearly normal $f_\theta(X)|Y$. 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 $10^2 - 10^4$.

Figures 1-3 show that in these cases of text and image data $f_\theta(X)|Y$ is approximately normal for both randomly drawn θ vectors (Figure 1) and for θ representing estimated classifiers (Figures 2 and 3). A caveat in this case is that normality may not hold when θ is sparse, as may happen for example for L_1 regularized models (last row of Figure 2).

From a theoretical standpoint normality may be argued using a central limit theorem. We examine below several progressingly 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 1 (de-Moivre)**.** *If $Z_i, i \in \mathbb{N}$ are iid with expectation μ and variance σ^2 and $\bar{Z}_d = d^{-1}\sum_{i=1}^d Z_i$ then we have the following convergence in distribution*

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

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 the data dimensions are often not iid.

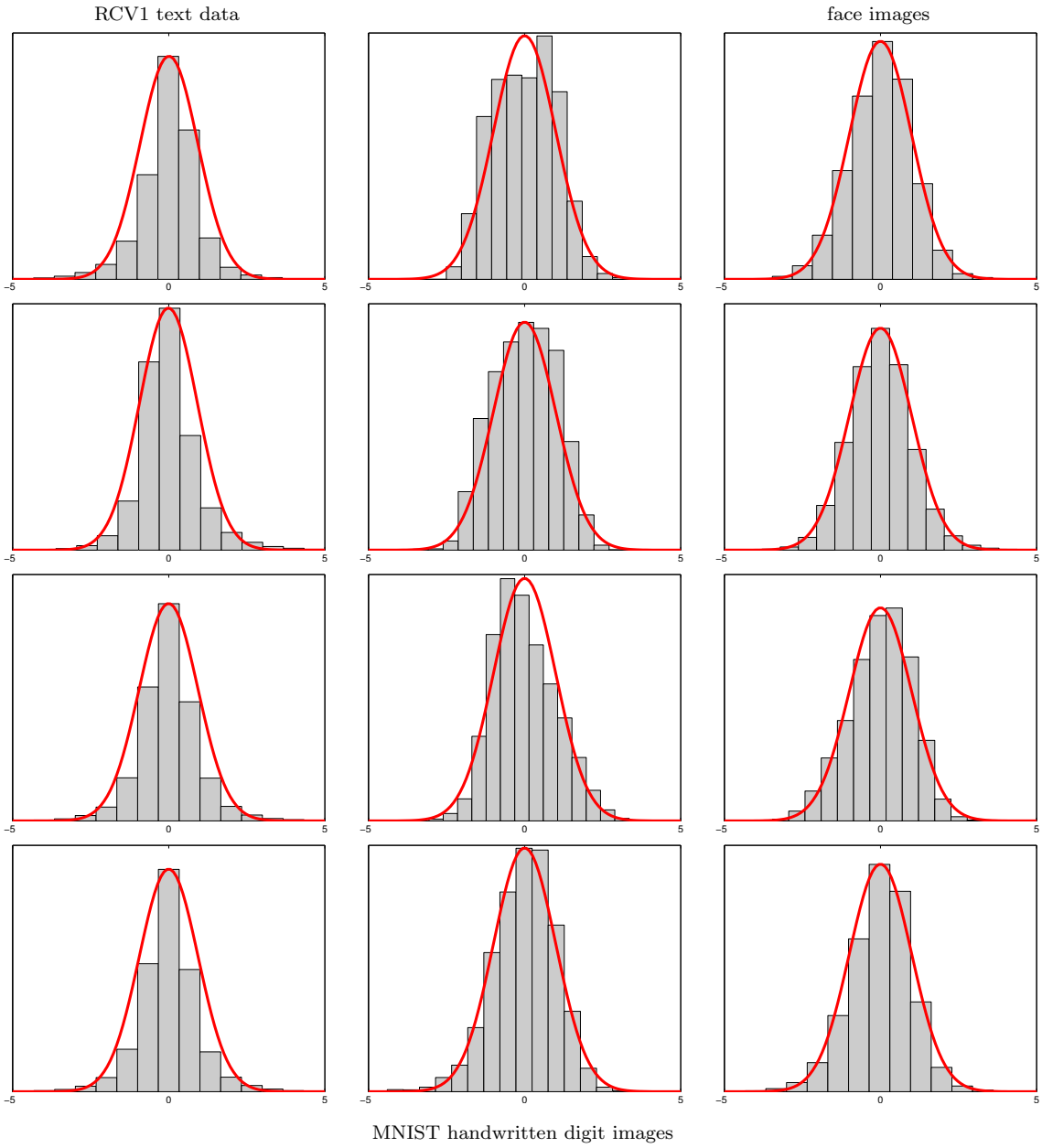


Figure 1: Centered histograms of $f_\theta(X)|\{Y = 1\}$ overlayed with the pdf of a fitted Gaussian for randomly drawn θ vectors ($\theta_i \sim U(-1/2, 1/2)$). The columns represent datasets (RCV1 text data [67], MNIST digit images, and face images [80]) and the rows represent multiple random draws. For uniformity we subtracted the empirical mean and divided by the empirical standard deviation. The twelve 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 often for randomly drawn θ.

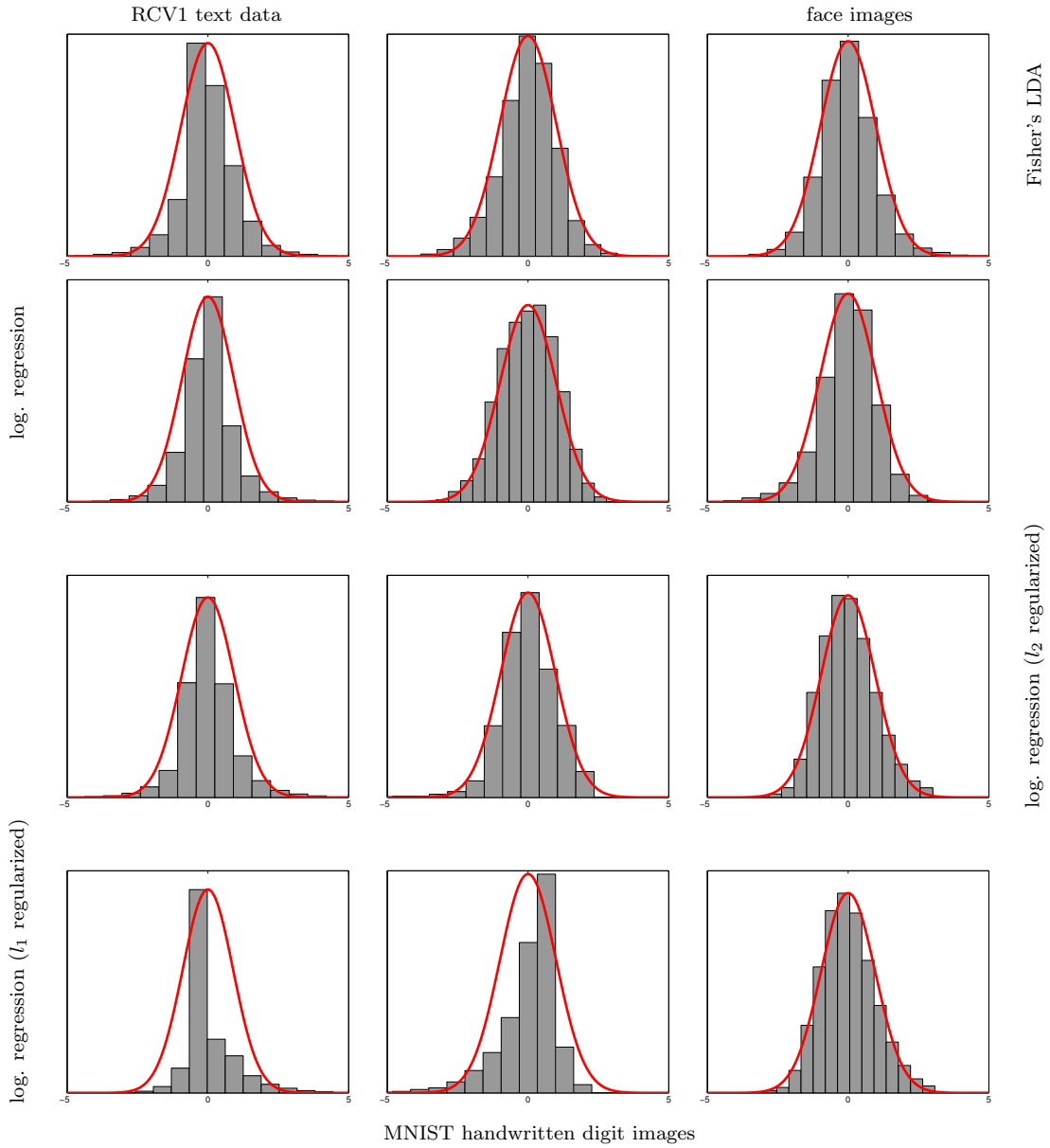


Figure 2: Centered histograms of $f_\theta(X)|\{Y = 1\}$ overlayed with the pdf of a fitted Gaussian for multiple θ vectors (four rows: Fisher's LDA, logistic regression, l_2 regularized logistic regression, and l_1 regularized logistic regression-all regularization parameters were selected by cross validation) and datasets (columns: RCV1 text data [67], MNIST digit images, and face images [80]). For uniformity we subtracted the empirical mean and divided by the empirical standard deviation. The twelve 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 for fitted θ values (except perhaps for L_1 regularization in the last row which promotes sparse θ).

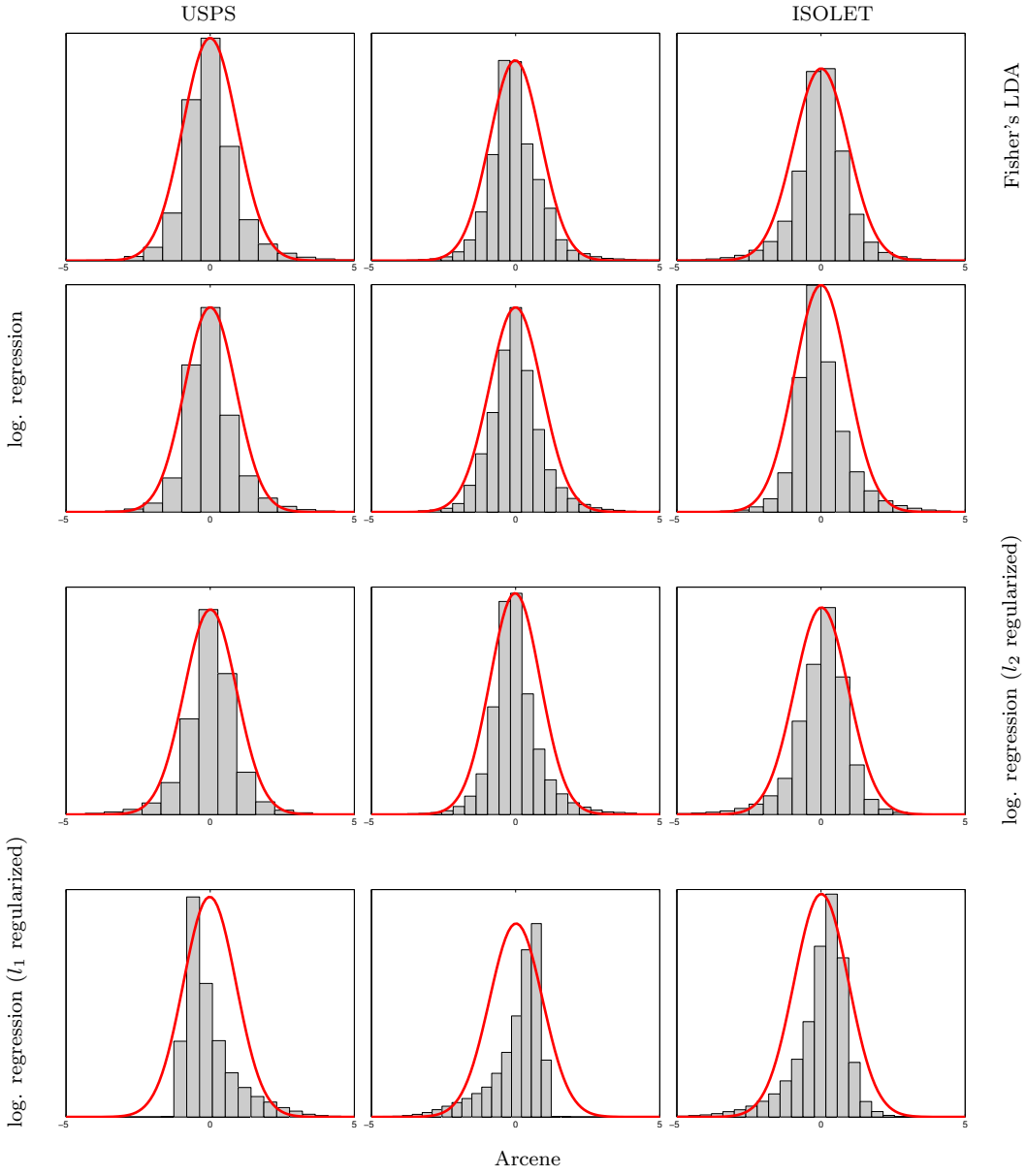


Figure 3: Centered histograms of $f_\theta(X)|\{Y = 1\}$ overlayed with the pdf of a fitted Gaussian for multiple θ vectors (four rows: Fisher's LDA, logistic regression, l_2 regularized logistic regression, and l_1 regularized logistic regression-all regularization parameters were selected by cross validation) and datasets (columns: USPS Handwritten Digits, Arcene data set, and ISOLET). For uniformity we subtracted the empirical mean and divided by the empirical standard deviation. The twelve panels further confirm that the assumption that $f_\theta(X)|Y$ is normal holds well for fitted θ values (except perhaps for L_1 regularization in the last row which promotes sparse θ) for various data sets.

A more general CLT does not require the summands Z_i to be identically distributed.

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

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

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

$$\lim_{d\to\infty} s_d^{-2} \sum_{i=1}^{d} \mathsf{E}(Z_i - \mu_i)^2 1_{\{|X_i - \mu_i| > \epsilon s_d\}} = 0. \tag{13}$$

This CLT is more general as it only requires that the data dimensions be independent. The condition (13) 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 inapplicable for dependent data dimensions.

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 $\{Z_1, \ldots, Z_r\}$, $\{Z_s, \ldots, Z_k\}$ are independent.

An early CLT for $m(k)$-dependent RVs was provided by [53]. Below is a slightly weakened version of the CLT, as proved in [11].

Proposition 3 (Berk)**.** *For each $k \in \mathbb{N}$ let $d(k)$ and $m(k)$ be increasing sequences and suppose that $Z_1^{(k)}, \ldots, Z_{d(k)}^{(k)}$ is an $m(k)$-dependent sequence of random variables. If*

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

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

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

Proposition 3 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 (in the sense of Definition 1). Intuitively, dimensions whose indices are far removed from each other are independent. The full power of Proposition 3 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.

A more realistic variation of $m(k)$ dependence where the dependency of each variable is specified using a dependency graph (rather than each dimension depends on neighboring dimensions) is advocated in a number of chapters, including the following recent result by [87].

Definition 2. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ indexing random variables is called a dependency graph if for any pair of disjoint subsets of $\mathcal{V}$, A_1 and A_2 such that no edge in $\mathcal{E}$ has one endpoint in A_1 and the other in A_2, we have independence between $\{Z_i : i \in A_1\}$ and $\{Z_i : i \in A_2\}$. The degree $d(v)$ of a vertex is the number of edges connected to it and the maximal degree is $\max_{v\in\mathcal{V}} d(v)$.

Proposition 4 (Rinott)**.** *Let* $Z_1, \ldots, Z_n$ *be random variables having a dependency graph whose maximal degree is strictly less than* D*, satisfying* $|Z_i - \mathsf{E}\, Z_i| \leq B$ *a.s.,* $\forall i$*,* $\mathsf{E}(\sum_{i=1}^n Z_i) = \lambda$ *and* $\mathsf{Var}(\sum_{i=1}^n Z_i) = \sigma^2 > 0$*, Then for any* $w \in \mathbb{R}$*,*

$$\left| P\left(\frac{\sum_{i=1}^n Z_i - \lambda}{\sigma} \leq w \right) - \Phi(w) \right| \leq \frac{1}{\sigma} \left(\frac{1}{\sqrt{2\pi}} DB + 16 \left(\frac{n}{\sigma^2} \right)^{1/2} D^{3/2} B^2 + 10 \left(\frac{n}{\sigma^2} \right) D^2 B^3 \right)$$

where $\Phi(w)$ *is the CDF corresponding to a N(0,1) distribution.*

The above theorem states a stronger result than convergence in distribution to a Gaussian in that it states a uniform rate of convergence of the CDF. Such results are known in the literature as Berry Essen bounds [31]. When D and B are bounded and $\mathsf{Var}\left(\sum_{i=1}^{n} Z_i\right) = O(n)$ it yields a CLT with an optimal convergence rate of $n^{-1/2}$.

The question of whether the above CLTs apply in practice is a delicate one. For text one can argue that the appearance of a word depends on some words but is independent of other words. Similarly for images it is plausible to say that the brightness of a pixel is independent of pixels that are spatially far removed from it. In practice one needs to verify the normality assumption empirically, which is simple to do by comparing the empirical histogram of $f_\theta(X)$ with that of a fitted mixture of Gaussians. As the figures above indicate this holds for text and image data for some values of θ, assuming it is not sparse. Also, it is worth mentioning that one dimensional CLTs kick in relatively early perhaps at 50 or 100 dimensions. Even when the high dimensional data lie on a lower dimensional manifold whose dimensionality is on the order of 100 dimensions, the CLT still applies to some extent (see histogram plots).

2.2.2 Unsupervised Consistency of $\hat{R}_n(\theta)$

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

Definition 3. A parametric family $\{p_\alpha : \alpha \in A\}$ is identifiable when $p_\alpha(x) = p_{\alpha'}(x), \forall x$ implies $\alpha = \alpha'$.

Proposition 5. *Assuming known label marginals with* $\mathbb{P}(Y = 1) \neq \mathbb{P}(Y = -1)$ *the*

Gaussian mixture family

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

is identifiable.

Proof. It can be shown that the family of Gaussian mixture model with no apriori information about label marginals is identifiable up to a permutation of the labels y [106]. We proceed by assuming with no loss of generality that $\mathbb{P}(Y=1) > \mathbb{P}(Y=-1)$. The alternative case $\mathbb{P}(Y=1) < \mathbb{P}(Y=-1)$ may be handled in the same manner. Using the result of [106] we have that if $p_{\mu,\sigma}(x) = p_{\mu',\sigma'}(x)$ for all x, then $(\mathbb{P}(y),\mu,\sigma) = (\mathbb{P}(y),\mu',\sigma')$ up to a permutation of the labels. Since permuting the labels violates our assumption $\mathbb{P}(Y=1) > \mathbb{P}(Y=-1)$ we establish $(\mu,\sigma) = (\mu',\sigma')$ proving identifiability. □

The assumption that $\mathbb{P}(Y)$ is known is not entirely crucial. It may be relaxed by assuming that it is known whether $\mathbb{P}(Y=1) > \mathbb{P}(Y=-1)$ or $\mathbb{P}(Y=1) < \mathbb{P}(Y=-1)$. Proving Proposition 5 under this much weaker assumption follows identical lines.

Proposition 6. *Under the assumptions of Proposition 5 the MLE estimates for* $(\mu,\sigma) = (\mu_1,\mu_{-1},\sigma_1,\sigma_{-1})$

$$(\hat{\mu}^{(n)},\hat{\sigma}^{(n)}) = \arg\max_{\mu,\sigma} \ell_n(\mu,\sigma)$$
$$\ell_n(\mu,\sigma) = \sum_{i=1}^{n} \log \sum_{y^{(i)}\in\{-1,+1\}} \mathbb{P}(y^{(i)})\mathbb{P}_{\mu_y,\sigma_y}(f_\theta(X^{(i)})|y^{(i)}).$$

are consistent i.e., $(\hat{\mu}_1^{(n)},\hat{\mu}_{-1}^{(n)},\hat{\sigma}_1^{(n)},\hat{\sigma}_{-1}^{(n)})$ *converge as* $n\to\infty$ *to the true parameter values with probability 1.*

Proof. Denoting $p_\eta(z) = \sum_y \mathbb{P}(y)\mathbb{P}_{\mu_y,\sigma_y}(z|y)$ with $\eta = (\mu,\sigma)$ we note that p_η is identifiable (see Proposition 5) in η and the available samples $z^{(i)} = f_\theta(X^{(i)})$ are iid samples from $p_\eta(z)$. We therefore use standard statistics theory which indicates that the MLE for identifiable parametric model is strongly consistent [43, chap. 17]. □

Proposition 7. *Under the assumptions of Proposition 5 and assuming the loss $\mathcal{L}$ is given by one of* (2)-(4) *with a normal $f_\theta(X)|Y \sim N(\mu_y, \sigma_y^2)$, the plug-in risk estimate*

$$\hat{R}_n(\theta) = \sum_{y\in\{-1,+1\}} \mathbb{P}(y) \int_{\mathbb{R}} \mathbb{P}_{\hat{\mu}_y^{(n)},\hat{\sigma}_y^{(n)}}(f_\theta(X) = \alpha|y)\mathcal{L}(y,\alpha)\, d\alpha. \tag{14}$$

is consistent, i.e., for all θ,

$$P\left(\lim_n \hat{R}_n(\theta) = R(\theta)\right) = 1.$$

Proof. The plug-in risk estimate $\hat{R}_n$ in (14) is a continuous function (when L is given by (2), (3) or (4)) of $\hat{\mu}_1^{(n)}, \hat{\mu}_{-1}^{(n)}, \hat{\sigma}_1^{(n)}, \hat{\sigma}_{-1}^{(n)}$ (note that μ_y and σ_y are functions of θ), 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 6 we have that

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

with probability 1. Since continuous functions preserve limits we have

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

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

2.2.3 Unsupervised Consistency of $\arg\min \hat{R}_n(\theta)$

The convergence above $\hat{R}_n(\theta) \to R(\theta)$ is pointwise in θ. If the stronger concept of uniform convergence is assumed over $\theta \in \Theta$ we obtain consistency of $\arg\min_\theta \hat{R}_n(\theta)$. This surprising result indicates that in some cases it is possible to retrieve the expected risk minimizer (and therefore the Bayes classifier in the case of the hinge loss, log-loss and exp-loss) using only unlabeled data. We show this uniform convergence using a modification of Wald's classical MLE consistency result [43, chap. 17].

Denoting

$$p_\eta(z) = \sum_{y\in\{-1,+1\}} \mathbb{P}(y)\mathbb{P}_{\mu_y,\sigma_y}(f(X) = z|y), \quad \eta = (\mu_1, \mu_{-1}, \sigma_1, \sigma_{-1})$$

we first show that the MLE converges to the true parameter value $\hat{\eta}_n \to \eta_0$ uniformly. Uniform convergence of the risk estimator $\hat{R}_n(\theta)$ follows. Since changing $\theta \in \Theta$ results in a different $\eta \in E$ we can state the uniform convergence in $\theta \in \Theta$ or alternatively in $\eta \in E$.

Proposition 8. *Let θ take values in Θ for which $\eta \in E$ for some compact set E. Then assuming the conditions in Proposition 7 the convergence of the MLE to the true value $\hat{\eta}_n \to \eta_0$ is uniform in $\eta_0 \in E$ (or alternatively $\theta \in \Theta$).*

Proof. We start by making the following notation

$$U(z, \eta, \eta_0) = \log p_\eta(z) - \log p_{\eta_0}(z)$$
$$\alpha(\eta, \eta_0) = E_{p_{\eta_0}} U(z, \eta, \eta_0) = -D(p_{\eta_0}, p_\eta) \leq 0$$

with the latter quantity being non-positive and 0 iff $\eta = \eta_0$ (due to Shannon's inequality and identifiability of p_η).

For $\rho > 0$ we define the compact set $S_{\eta_0,\rho} = \{\eta \in E : \|\eta - \eta_0\| \geq \rho\}$. Since $\alpha(\eta, \eta_0)$ is continuous it achieves its maximum (with respect to η) on $S_{\eta_0,\rho}$ denoted by $\delta_\rho(\eta_0) = \max_{\eta \in S_{\eta_0,\rho}} \alpha(\eta, \eta_0) < 0$ which is negative since $\alpha(\eta, \eta_0) = 0$ iff $\eta = \eta_0$. Furthermore, note that $\delta_\rho(\eta_0)$ is itself continuous in $\eta_0 \in E$ and since E is compact it achieves its maximum

$$\delta = \max_{\eta_0 \in E} \delta_\rho(\eta_0) = \max_{\eta_0 \in E} \max_{\eta \in S_{\eta_0,\rho}} \alpha(\eta, \eta_0) < 0$$

which is negative for the same reason.

Invoking the uniform strong law of large numbers [43, chap. 16] we have

$$n^{-1} \sum_{i=1}^{n} U(z^{(i)}, \eta, \eta_0) \to \alpha(\eta, \eta_0)$$

uniformly over $(\eta, \eta_0) \in E^2$. Consequentially, there exists N such that for $n > N$ (with probability 1)

$$\sup_{\eta_0 \in E} \sup_{\eta \in S_{\eta_0,\rho}} \frac{1}{n} \sum_{i=1}^{n} U(z^{(i)}, \eta, \eta_0) < \delta/2 < 0.$$

But since $n^{-1}\sum_{i=1}^{n} U(z^{(i)},\eta,\eta_0) \to 0$ for $\eta=\eta_0$ it follows that the MLE

$$\hat{\eta}_n = \max_{\eta\in E} \frac{1}{n}\sum_{i=1}^{n} U(z^{(i)},\eta,\eta_0)$$

is outside $S_{\eta_0,\rho}$ (for $n>N$ uniformly in $\eta_0\in E$) which implies $\|\hat{\eta}_n-\eta_0\|\leq\rho$. Since $\rho>0$ is arbitrarily and N does not depend on η_0 we have $\hat{\eta}_n\to\eta_0$ uniformly over $\eta_0\in E$. □

Proposition 9. *Assuming that X,Θ are bounded in addition to the assumptions of Proposition 8 the convergence $\hat{R}_n(\theta)\to R(\theta)$ is uniform in $\theta\in\Theta$.*

Proof. Since X,Θ are bounded the margin value $f_\theta(X)$ is bounded with probability 1. As a result the loss function is bounded in absolute value by a constant C. We also note that a mixture of two Gaussian model (with known mixing proportions) is Lipschitz continuous in its parameters

$$\left|\sum_{y\in\{-1,+1\}} \mathbb{P}(y)\mathbb{P}_{\hat{\mu}_y^{(n)},\hat{\sigma}_y^{(n)}}(z) - \sum_{y\in\{-1,+1\}} \mathbb{P}(y)\mathbb{P}_{\mu_y^{true},\sigma_y^{true}}(z)\right| \leq t(z)\cdot\left|\left|(\hat{\mu}_1^{(n)},\hat{\mu}_{-1}^{(n)},\hat{\sigma}_1^{(n)},\hat{\sigma}_{-1}^{(n)}) - (\mu_1^{\text{true}},\mu_{-1}^{\text{true}},\sigma_1^{\text{true}},\sigma_{-1}^{\text{true}})\right|\right|$$

which may be verified by noting that the partial derivatives of $p_\eta(z)=\sum_y \mathbb{P}(y)p_{\mu_y,\sigma_y}(z|y)$

$$\frac{\partial p_\eta(z)}{\partial\hat{\mu}_1^{(n)}} = \frac{\mathbb{P}(Y=1)(z-\hat{\mu}_1^{(n)})}{(2\pi)^{1/2}\hat{\sigma}_1^{(n)^3}} e^{-\frac{(z-\hat{\mu}_1^{(n)})^2}{2\hat{\sigma}_1^{(n)^3}}}$$

$$\frac{\partial p_\eta(z)}{\partial\hat{\mu}_{-1}^{(n)}} = \frac{\mathbb{P}(Y=-1)(z-\hat{\mu}_{-1}^{(n)})}{(2\pi)^{1/2}\hat{\sigma}_{-1}^{(n)^3}} e^{-\frac{(z-\hat{\mu}_{-1}^{(n)})^2}{2\hat{\sigma}_{-1}^{(n)^3}}}$$

$$\frac{\partial p_\eta(z)}{\partial\hat{\sigma}_1^{(n)}} = -\frac{\mathbb{P}(Y=1)(z-\hat{\mu}_1^{(n)})^2}{(2\pi)^{3/2}\hat{\sigma}_1^{(n)^6}} e^{-\frac{(z-\hat{\mu}_1^{(n)})^2}{2\hat{\sigma}_1^{(n)^2}}}$$

$$\frac{\partial p_\eta(z)}{\partial\hat{\sigma}_{-1}^{(n)}} = -\frac{\mathbb{P}(Y=-1)(z-\hat{\mu}_{-1}^{(n)})^2}{(2\pi)^{3/2}\hat{\sigma}_{-1}^{(n)^6}} e^{-\frac{(z-\hat{\mu}_{-1}^{(n)})^2}{2\hat{\sigma}_{-1}^{(n)^2}}}$$

are bounded for a compact E. These observations, together with Proposition 8 lead to

$$\begin{aligned}
|\hat{R}_n(\theta) - R(\theta)| &\leq \sum_{y\in\{-1,+1\}} \mathbb{P}(y) \int \left| \mathbb{P}_{\hat{\mu}_y^{(n)},\hat{\sigma}_y^{(n)}}(f_\theta(X)=\alpha) - \mathbb{P}_{\mu_y^{\text{true}},\sigma_y^{\text{true}}}(f_\theta(X)=\alpha) \right| |\mathcal{L}(y,\alpha)| d\alpha \\
&\leq C \int \Big| \sum_{y\in\{-1,+1\}} \mathbb{P}(y)\mathbb{P}_{\hat{\mu}_y^{(n)},\hat{\sigma}_y^{(n)}}(\alpha) - \sum_{y\in\{-1,+1\}} \mathbb{P}(y)\mathbb{P}_{\mu_y^{\text{true}},\sigma_y^{\text{true}}}(\alpha) \Big| d\alpha \\
&\leq C \left\| (\hat{\mu}_1^{(n)}, \hat{\mu}_{-1}^{(n)}, \hat{\sigma}_1^{(n)}, \hat{\sigma}_{-1}^{(n)}) - (\mu_1^{\text{true}}, \mu_{-1}^{\text{true}}, \sigma_1^{\text{true}}, \sigma_{-1}^{\text{true}}) \right\| \int_a^b t(z)dz \\
&\leq C' \left\| (\hat{\mu}_1^{(n)}, \hat{\mu}_{-1}^{(n)}, \hat{\sigma}_1^{(n)}, \hat{\sigma}_{-1}^{(n)}) - (\mu_1^{\text{true}}, \mu_{-1}^{\text{true}}, \sigma_1^{\text{true}}, \sigma_{-1}^{\text{true}}) \right\| \to 0
\end{aligned}$$

uniformly over $\theta \in \Theta$. □

Proposition 10. *Under the assumptions of Proposition 9*

$$P\left(\lim_{n\to\infty} \arg\min_{\theta\in\Theta} \hat{R}_n(\theta) = \arg\min_{\theta\in\Theta} R(\theta) \right) = 1.$$

Proof. We denote $t^* = \arg\min R(\theta)$, $t_n = \arg\min \hat{R}_n(\theta)$. Since $\hat{R}_n(\theta) \to R(\theta)$ uniformly, for each $\epsilon > 0$ there exists N such that for all $n > N$, $|\hat{R}_n(\theta) - R(\theta)| < \epsilon$.

Let $S = \{\theta : \|\theta - t^*\| \geq \epsilon\}$ and $\min_{\theta\in S} R(\theta) > R(t^*)$ (S is compact and thus R achieves its minimum on it). There exists N' such that for all $n > N'$ and $\theta \in S$, $\hat{R}_n(\theta) \geq R(t^*) + \epsilon$. On the other hand, $\hat{R}_n(t^*) \to R(t^*)$ which together with the previous statement implies that there exists N'' such that for $n > N''$, $\hat{R}_n(t^*) < \hat{R}_n(\theta)$ for all $\theta \in S$. We thus conclude that for $n > N''$, $t_n \notin S$. Since we showed that for each $\epsilon > 0$ there exists N such that for all $n > N$ we have $\|t_n - t^*\| \leq \epsilon$, $t_n \to t^*$ which concludes the proof. □

2.2.4 Asymptotic Variance

In addition to consistency, it is useful to characterize the accuracy of our estimator $\hat{R}_n(\theta)$ as a function of $\mathbb{P}(y), \mu, \sigma$. We do so by computing the asymptotic variance of

the estimator which equals the inverse Fisher information

$$\sqrt{n}(\hat{\eta}_n^{\text{mle}} - \eta_0) \rightsquigarrow N(0, I^{-1}(\eta^{\text{true}}))$$

and analyzing its dependency on the model parameters. We first derive the asymptotic variance of MLE for mixture of Gaussians (we denote below $\eta = (\eta_1, \eta_2), \eta_i = (\mu_i, \sigma_i)$)

$$\begin{aligned} p_\eta(z) &= \mathbb{P}(Y=1)N(z;\mu_1,\sigma_1^2) + \mathbb{P}(Y=-1)N(z;\mu_{-1},\sigma_{-1}^2) \\ &= p_1 p_{\eta_1}(z) + p_{-1} p_{\eta_{-1}}(z). \end{aligned}$$

The elements of 4×4 information matrix $I(\eta)$

$$I(\eta_i, \eta_j) = \mathsf{E}\left(\frac{\partial \log p_\eta(z)}{\partial \eta_i} \frac{\partial \log p_\eta(z)}{\partial \eta_j}\right)$$

may be computed using the following derivatives

$$\begin{aligned} \frac{\partial \log p_\eta(z)}{\partial \mu_i} &= \frac{p_i}{\sigma_i}\left(\frac{z-\mu_i}{\sigma_i}\right)\frac{p_{\eta_i}(z)}{p_\eta(z)} \\ \frac{\partial \log p_\eta(z)}{\partial \sigma_i^2} &= \frac{p_i}{2\sigma_i}\left(\left(\frac{z-\mu_i}{\sigma_i}\right)^2 - 1\right)\frac{p_{\eta_i}(z)}{p_\eta(z)} \end{aligned}$$

for $i = 1, -1$. Using the method of [9] we obtain

$$\begin{aligned} I(\mu_i, \mu_j) &= \frac{p_i p_j}{\sigma_i \sigma_j} M_{11}\Big(p_{\eta_i}(z), p_{\eta_i}(z)\Big) \\ I(\mu_1, \sigma_i^2) &= \frac{p_1 p_i}{2\sigma_1 \sigma_i^2}\Big[M_{12}\Big(p_{\eta_i}(z), p_{\eta_i}(z)\Big) - M_{10}\Big(p_{\eta_1}(z), p_{\eta_i}(z)\Big)\Big] \\ I(\mu_{-1}, \sigma_i^2) &= \frac{p_{-1} p_i}{2\sigma_{-1} \sigma_i^2}\Big[M_{21}\Big(p_{\eta_i}(z), p_{\eta_{-1}}(z)\Big) - M_{01}\Big(p_{\eta_i}(z), p_{\eta_{-1}}(z)\Big)\Big] \\ I(\sigma_i^2, \sigma_i^2) &= \frac{p_i^4}{4\sigma_i^4}\Big[M_{00}\Big(p_{\eta_i}(z), p_{\eta_i}(z)\Big) - 2M_{11}\Big(p_{\eta_i}(z), p_{\eta_i}(z)\Big) + M_{22}\Big(p_{\eta_i}(z), p_{\eta_i}(z)\Big)\Big] \\ I(\sigma_1^2, \sigma_{-1}^2) &= \frac{p_1 p_{-1}}{4\sigma_1^2 \sigma_{-1}^2}\Big[M_{00}\Big(p_{\eta_1}(z), p_{\eta_{-1}}(z)\Big) - M_{20}\Big(p_{\eta_1}(z), p_{\eta_{-1}}(z)\Big) \\ &\qquad - M_{02}\Big(p_{\eta_1}(z), p_{\eta_{-1}}(z)\Big) + M_{22}\Big(p_{\eta_1}(z), p_{\eta_{-1}}(z)\Big)\Big] \end{aligned}$$

where

$$M_{m,n}\Big(p_{\eta_i}(z), p_{\eta_j}(z)\Big) = \int_{-\infty}^{\infty}\left(\frac{z-\mu_i}{\sigma_i}\right)^m\left(\frac{z-\mu_j}{\sigma_j}\right)^n \frac{p_{\eta_i}(z)p_{\eta_j}(z)}{p_\eta(z)}\,dx.$$

In some cases it is more instructive to consider the asymptotic variance of the risk estimator $\hat{R}_n(\theta)$ rather than that of the parameter estimate for $\eta = (\mu, \sigma)$. This could be computed using the delta method and the above Fisher information matrix

$$\sqrt{n}(\hat{R}_n(\theta) - R(\theta)) \rightsquigarrow N(0, \nabla h(\eta^{\text{true}})^T I^{-1}(\eta^{true}) \nabla h(\eta^{\text{true}}))$$

where ∇h is the gradient vector of the mapping $R(\theta) = h(\eta)$. For example, in the case of the exponential loss (2) we get

$$h(\eta) = \mathbb{P}(Y=1)\sigma_1\sqrt{2}\exp\Big(\frac{(\mu_1-1)^2}{2} - \frac{\mu_1^2}{2\sigma_1^2}\Big) + \mathbb{P}(Y=-1)\sigma_{-1}\sqrt{2}\exp\Big(\frac{(\mu_{-1}-1)^2}{2} - \frac{\mu_{-1}^2}{2\sigma_{-1}^2}\Big)$$

$$\frac{\partial h(\eta)}{\partial \mu_1} = \frac{\sqrt{2}\mathbb{P}(Y=1)(\mu_1(\sigma_1^2-1)-\sigma_1^2)}{\sigma_1}\exp\left(\frac{(\mu_1-1)^2}{2} - \frac{\mu_1^2}{2\sigma_1^2}\right)$$

$$\frac{\partial h(\eta)}{\partial \mu_{-1}} = \frac{\sqrt{2}\mathbb{P}(Y=-1)(\mu_{-1}(\sigma_{-1}^2-1)+\sigma_{-1}^2)}{\sigma_{-1}}\exp\left(\frac{(\mu_{-1}+1)^2}{2} - \frac{\mu_{-1}^2}{2\sigma_{-1}^2}\right)$$

$$\frac{\partial h(\eta)}{\partial \sigma_1^2} = \frac{\mathbb{P}(Y=1)(\mu_1^2+\sigma_1^2)}{\sqrt{2\sigma_1}}\Big(\frac{(\mu_1-1)^2}{2} - \frac{\mu_1^2}{2\sigma_1^2}\Big)$$

$$\frac{\partial h(\eta)}{\partial \sigma_{-1}^2} = \frac{\mathbb{P}(Y=-1)(\mu_{-1}^2+\sigma_{-1}^2)}{\sqrt{2\sigma_{-1}}}\Big(\frac{(\mu_{-1}+1)^2}{2} - \frac{\mu_{-1}^2}{2\sigma_{-1}^2}\Big).$$

Figure 4 plots the asymptotic accuracy of $\hat{R}_n(\theta)$ for log-loss. The left panel shows that the accuracy of $\hat{R}_n$ increases with the imbalance of the marginal distribution $\mathbb{P}(Y)$. The right panel shows that the accuracy of $\hat{R}_n$ increases with the difference between the means $|\mu_1 - \mu_{-1}|$ and the variances σ_1/σ_2.

2.2.5 Multiclass Classification

Thus far, we have considered unsupervised risk estimation in binary classification. In this section we describe a multiclass extension based on standard extensions of the margin concept to multiclass classification. In this case the margin vector associated with the multiclass classifier

$$\hat{Y} = \underset{k=1,\ldots,K}{\arg\max}\, f_{\theta^k}(X), \qquad X, \theta^k \in \mathbb{R}^d$$

is $f_{\boldsymbol{\theta}}(X) = (f_{\theta^1}(X), \ldots, f_{\theta^K}(X))$. Following our discussion of the binary case, $f_{\theta^k}(X)|Y$, $k = 1, \ldots, K$ is assumed to be normally distributed with parameters that are estimated by maximizing the likelihood of a Gaussian mixture model. We thus have

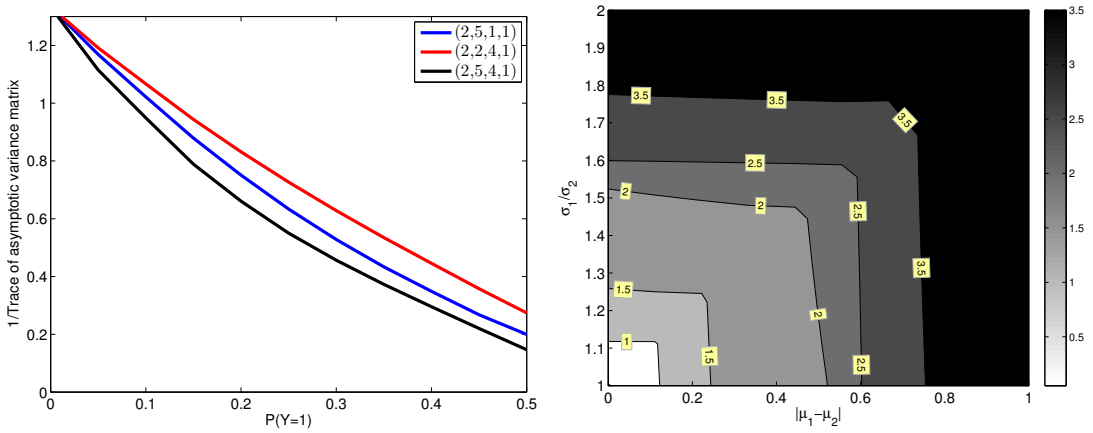


Figure 4: Left panel: asymptotic accuracy (inverse of trace of asymptotic variance) of $\hat{R}_n(\theta)$ for logloss as a function of the imbalance of the class marginal $\mathbb{P}(Y)$. The accuracy increases with the class imbalance as it is easier to separate the two mixture components. Right panel: asymptotic accuracy (inverse of trace of asymptotic variance) as a function of the difference between the means $|\mu_1 - \mu_{-1}|$ and the variances σ_1/σ_2. See text for more information.

K Gaussian mixture models, each one with K mixture components. The estimated parameters are plugged-in as before into the multiclass risk

$$R(\boldsymbol{\theta}) = E_{p(f_{\boldsymbol{\theta}}(X),Y)}\mathcal{L}(Y, f_{\boldsymbol{\theta}}(X))$$

where $\mathcal{L}$ is a multiclass margin based loss function such as

$$\mathcal{L}(Y, f_{\boldsymbol{\theta}}(X)) = \sum_{k \neq Y} \log(1 + \exp(-f_{\theta^k}(X))) \tag{15}$$

$$\mathcal{L}(Y, f_{\boldsymbol{\theta}}(X)) = \sum_{k \neq Y} (1 + f_{\theta^k}(X))_+. \tag{16}$$

Care should be taken when defining the loss function for the multi-class case, as a stright-forward extension from the binary case might render the framework inconsistent. We use the specific extension which is proved to be consistent for various loss functions (including hinge-loss) by [107]. Since the MLE for a Gaussian mixture model with K components is consistent (assuming $\mathbb{P}(Y)$ is known and all probabilities $\mathbb{P}(Y = k), k = 1, \ldots, K$ are distinct) the MLE estimator for $f_{\theta^k}(X)|Y = k'$ are consistent. Furthermore, if the loss $\mathcal{L}$ is a continuous function of these parameters (as is the case for (15)-(16)) the risk estimator $\hat{R}_n(\boldsymbol{\theta})$ is consistent as well.

2.3 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 $\mathbb{P}(Y)$ and classification accuracy. This controlled setting allows us to examine the accuracy of the risk estimator as a function of n, $\mathbb{P}(Y)$, and the classifier accuracy.

Figure 5 shows that the relative error of $\hat{R}_n(\theta)$ (measured by $|\hat{R}_n(\theta)-R_n(\theta)|/R_n(\theta)$) in estimating the logloss (left) and hinge loss (right). The curves decrease with n and achieve accuracy of greater than 99% for $n > 1000$. In accordance with the theoretical results in Section 2.2.4 the figure shows that the estimation error decreases as the classifiers become more accurate and as $\mathbb{P}(Y)$ becomes less uniform. We found these trends to hold in other experiments as well. In the case of exponential loss, however, the estimator performed substantially worse across the board, in some cases with an absolute error of as high as 10. This is likely due to the exponential dependency of the loss on $Yf_\theta(X)$ which makes it very sensitive to outliers.

Figure 23 shows the accuracy of logloss estimation for a real world transfer learning experiment based on the 20-newsgroup data. We followed the experimental setup of used by [30] in order to have different distributions for training and test sets. More specifically, 20-newsgroup data has a hierarchical class taxonomy and the transfer learning problem is defined at the top-level categories. We split the data based

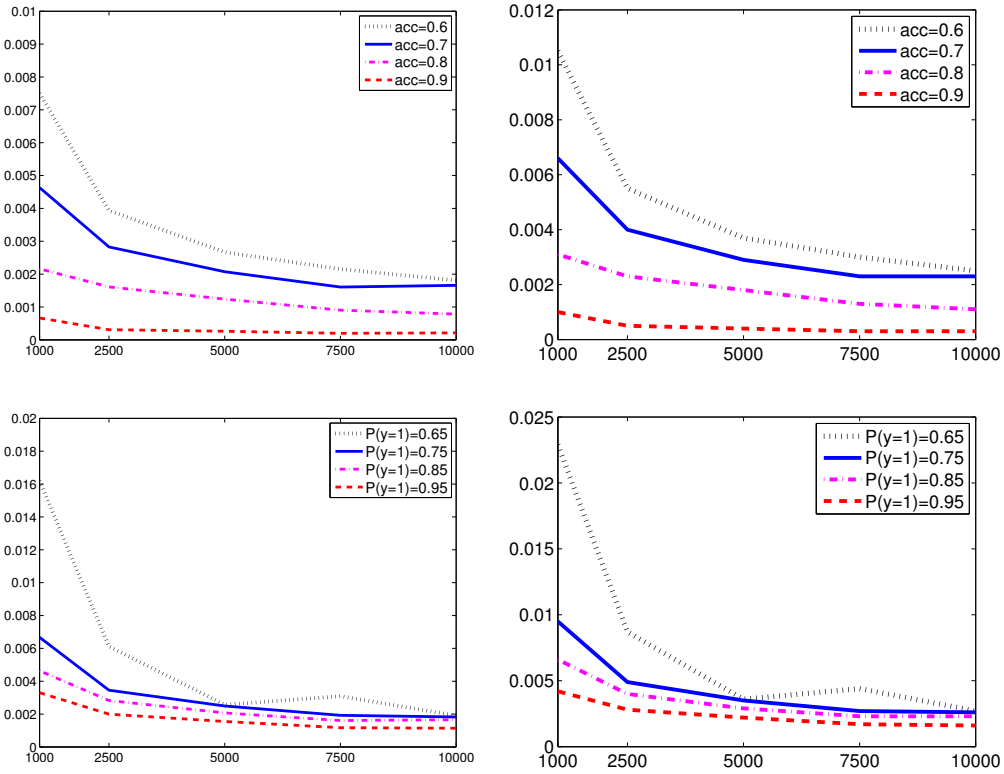


Figure 5: The relative accuracy of $\hat{R}_n$ (measured by $|\hat{R}_n(\theta) - R_n(\theta)|/R_n(\theta)$) as a function of n, classifier accuracy (acc) and the label marginal $\mathbb{P}(Y)$ (left: logloss, right: hinge-loss). The estimation error nicely decreases with n (approaching 1% at $n = 1000$ and decaying further). It also decreases with the accuracy of the classifier (top) and non-uniformity of $\mathbb{P}(Y)$ (bottom) in accordance with the theory of Section 2.2.4.

Data	R_n	$\lvert R_n - \hat{R}_n \rvert$	$\lvert R_n - \hat{R}_n \rvert / R_n$	n	$\mathbb{P}(Y=1)$
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 6: 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 by [30] which may be referred to for more detail. The train and testing sets contained samples from two top categories in the topic hierarchy but with different subcategory proportions. The first column indicates the top category classification task and the second indicates the empirical log-loss R_n calculated using the true labels of the testing set (5). The third and forth columns indicate the absolute and relative errors of $\hat{R}_n$. The fifth and sixth columns indicate the train set size and the label marginal distribution.

on subcategories such that the training and test sets contain data sampled from different subcategories within the same top-level category. Hence, the training and test distributions differ. We trained a logistic regression classifier on the training set and estimate its risk on the test set of a different distribution. Our unsupervised risk estimator was quite effective in estimating the risk with relative accuracy greater than 96% and absolute error less than 0.02.

2.4 Application 2: Unsupervised Learning of Classifiers

Our second application is a very ambitious one: training classifiers using unlabeled data by minimizing the unsupervised risk estimate $\hat{\theta}_n = \arg\min \hat{R}_n(\theta)$. We evaluate the performance of the learned classifier $\hat{\theta}_n$ based on three quantities: (i) the unsupervised risk estimate $\hat{R}_n(\hat{\theta}_n)$, (ii) the supervised risk estimate $R_n(\hat{\theta}_n)$, and (iii) its classification error rate. We also compare the performance of $\hat{\theta}_n = \arg\min \hat{R}_n(\theta)$ with that of its supervised analog $\arg\min R_n(\theta)$.

We compute $\hat{\theta}_n = \arg\min \hat{R}_n(\theta)$ using two algorithms (see Algorithms 1-2) that start with an initial $\theta^{(0)}$ and iteratively construct a sequence of classifiers $\theta^{(1)}, \ldots, \theta^{(T)}$

which steadily decrease $\hat{R}_n$. 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 (17). We compute the integral in the loss function estimator using numeric integration. Since the integral is one dimensional a variety of numeric methods may be used with high accuracy and fast computation. 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 $\hat{R}_n$ (iteratively optimize one dimension at a time). This amounts to greedy search converging to local maxima. The same might hold for Algorithm 1, but we observe that Algorithm 1 workds slightly better in practice, leading to lower test error with less number of training iterations.

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 by minimizing the unsupervised hinge-loss estimate.

Algorithm 1 Unsupervised Gradient Descent

Input: $X^{(1)}, \ldots, X^{(n)} \in \mathbb{R}^d$, $\mathbb{P}(Y)$, step size α
Initialize $t = 0$, $\theta^{(t)} = \theta^0 \in \mathbb{R}^d$
repeat
  Compute $f_{\theta^{(t)}}(X^{(j)}) = \langle \theta^{(t)}, X^{(j)} \rangle \ \forall j = 1, \ldots, n$
  Estimate $(\hat{\mu}_1, \hat{\mu}_{-1}, \hat{\sigma}_1, \hat{\sigma}_{-1})$ by maximizing (11)
  for $i = 1$ **to** d **do**
    Plug-in the estimates into (14) to approximate

$$\frac{\partial \hat{R}_n(\theta^{(t)})}{\partial \theta_i} = \frac{\hat{R}_n(\theta^{(t)} + h_i e_i) - \hat{R}_n(\theta^{(t)} - h_i e_i)}{2h_i}$$
$$(e_i \text{ is an all zero vector except for } [e_i]_i = 1) \tag{17}$$

  end for
  Form $\nabla \hat{R}_n(\theta^{(t)}) = (\frac{\partial \hat{R}_n(\theta^{(t)})}{\partial \theta_1^{(t)}}, \ldots, \frac{\partial \hat{R}_n(\theta^{(t)})}{\partial \theta_d^{(t)}})$
  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: $X^{(1)}, \ldots, X^{(n)} \in \mathbb{R}^d$, $\mathbb{P}(Y)$, grid-size τ
Initialize $\theta_i \sim \text{Uniform}(-2, 2)$ for all i
repeat
  for $i = 1$ **to** d **do**
    Construct τ points grid in the range $[\theta_i - 4\tau, \theta_i + 4\tau]$
    Compute the risk estimate (14) 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 (14)
  end for
until convergence
Output: linear classifier $\theta^{\text{final}} = \theta$

Figures 7-8 display $\hat{R}_n(\hat{\theta}_n)$, $R_n(\hat{\theta}_n)$ and error-rate$(\hat{\theta}_n)$ on the training and testing sets as on two real world datasets: RCV1 (text documents) and MNIST (handwritten digit images) 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 $\mathbb{P}(Y = 1) = 0.3$ and $n = 199328$. 70% of the data was chosen as a (unlabeled) training set and the rest was held-out as a test-set. 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 $\mathbb{P}(Y = 1) = 0.53$. We randomly choose 70% of the data as a training set and kept the rest as a testing set.

Figures 7-8 indicate that minimizing the unsupervised logloss estimate is quite effective in learning an accurate classifier without labels. Both the unsupervised and supervised risk estimates $\hat{R}_n(\hat{\theta}_n)$, $R_n(\hat{\theta}_n)$ decay nicely when computed over the train set as well as the test set. Also interesting is the decay of the error rate. For comparison purposes supervised logistic regression with the same n achieved only slightly better test set error rate: 0.05 on RCV1 (instead of 0.1) and 0.07 or MNIST (instead of 0.1).

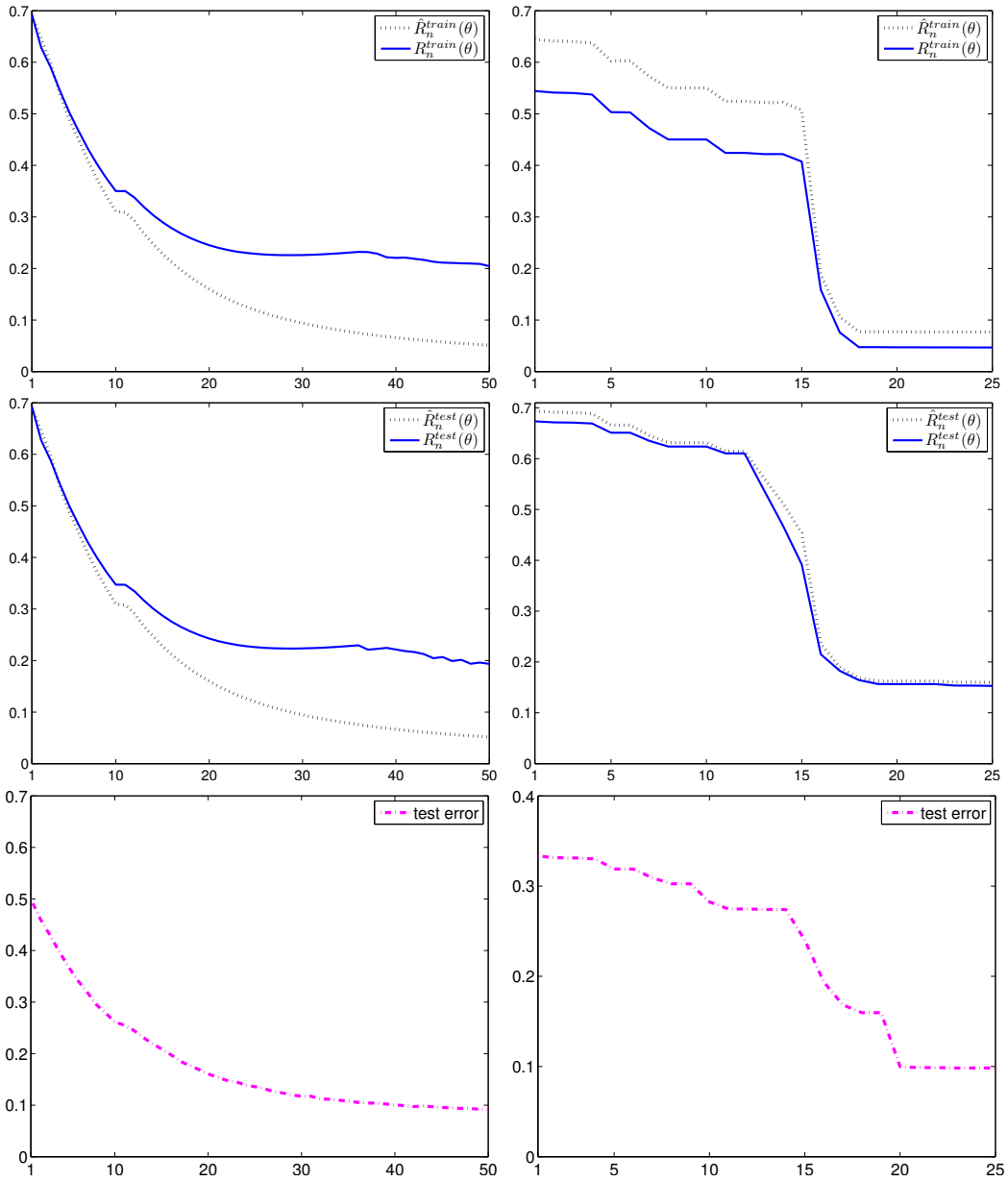


Figure 7: Performance of unsupervised logistic regression classifier $\hat{\theta}_n$ computed using Algorithm 1 (left) and Algorithm 2 (right) on the RCV1 dataset. The top two rows show the decay of the two risk estimates $\hat{R}_n(\hat{\theta}_n)$, $R_n(\hat{\theta}_n)$ as a function of the algorithm iterations. The risk estimates of $\hat{\theta}_n$ were computed using the train set (top) and the test set (middle). The bottom row displays the decay of the test set error rate of $\hat{\theta}_n$ as a function of the algorithm iterations. The figure shows that the algorithm obtains a relatively accurate classifier (testing set error rate 0.1, and $\hat{R}_n$ decaying similarly to R_n) without the use of a single labeled example. For comparison, the test error rate for supervised logistic regression with the same n is 0.07.

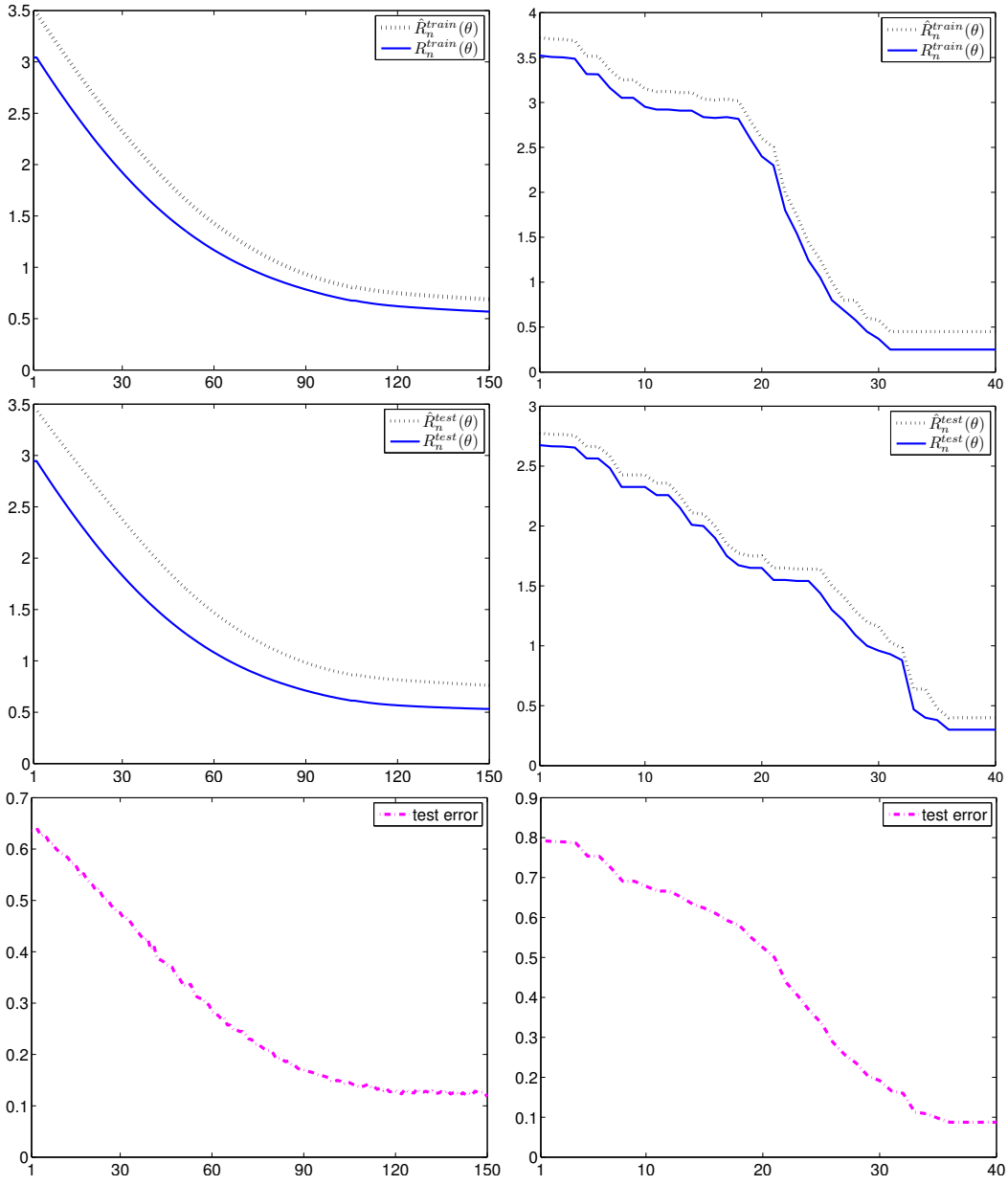


Figure 8: Performance of unsupervised logistic regression classifier $\hat{\theta}_n$ computed using Algorithm 1 (left) and Algorithm 2 (right) on the MNIST dataset. The top two rows show the decay of the two risk estimates $\hat{R}_n(\hat{\theta}_n)$, $R_n(\hat{\theta}_n)$ as a function of the algorithm iterations. The risk estimates of $\hat{\theta}_n$ were computed using the train set (top) and the test set (middle). The bottom row displays the decay of the test set error rate of $\hat{\theta}_n$ as a function of the algorithm iterations. The figure shows that the algorithm obtains a relatively accurate classifier (testing set error rate 0.1, and $\hat{R}_n$ decaying similarly to R_n) without the use of a single labeled example. For comparison, the test error rate for supervised logistic regression with the same n is 0.05.

In another experiment we examined the proposed approach on several different data sets and compared the classification performance with a supervised baseline (logistic regression) and Gaussian mixture modeling (GMM) clustering with known label proportions in the original data space (Table 1). The comparison was made under the same experimental setting (n, $\mathbb{P}(Y)$) for all three approaches. We used data sets from UCI machine learning repository [44] and from previously cited sources, unless otherwise noted. The following tasks were considered for each data set.

- RCV1: top category versus next 4 categories
- MNIST: Digit 1 versus Digit 2
- 20 newsgroups: Comp category versus Recreation category
- USPS: Digit 2 versus Digit 5
- Umist[0]: Male face (16 subjects) versus Female faces (4 subjects) with image resolution reduced to 40×40
- Arcene: Cancer versus Normal
- Isolet: Vowels versus Consonants
- Dexter: Documents about corporate acquisitions versus rest
- Secom: Semiconductor manufacturing defects versus good items
- Pham faces: Face versuss Non-face images
- CMU pie face: male (30 subjects) vs female (17 subjects)
- Madelon: It consists of data points (artificially generated) grouped in 32 clusters placed on the vertices of a five dimensional hypercube and randomly labeled +1 or -1, corrupted with features that are not useful for classification.

Table 1: Comparison (test set error rate) between supervised logistic regression, Unsupervised logistic regression and Gaussian mixture modeling in original data space. The unsupervised classifier performs better than the GMM clustering on the original space and compares well with its supervised counterpart on most data sets. See text for more details. The stars represent GMM with covariance $\sigma^2 I$ due to the high dimensionality. In all other cases we used a diagonal covariance matrix. Non-diagonal covariance matrix was impractical due to the high dimensionality.

Data set	Dimensions	Supervised log-reg	USL-2	GMM
RCV1	top 504 words	0.0500	0.0923	0.2083
Mnist	784	0.0700	0.1023	0.3163
20 news group	top 750 words	0.0652	0.0864	0.1234
USPS	256	0.0348	0.0545	0.1038
Umist	400 PCA components	0.1223	0.1955	0.2569
Arcene	1000 PCA components	0.1593	0.1877	0.3843*
Isolet	617	0.0462	0.0568	0.1332
Dexter	top-700 words	0.0564	0.1865	0.2715
Secom	591	0.1246	0.1532	0.2674
Pham faces	400	0.1157	0.1669	0.2324
CMU pie face	1024	0.0983	0.1386	0.2682*
Madelon	500	0.0803	0.1023	0.1120

Table 1 displays the test set error for the three methods on each data set. We note that our unsupervised approach achieves test set errors comparable to the supervised logistic regression in several data sets. The poor performance of the unsupervised technique on the Dexter data set is due to the fact that the data contains many irrelevant features. In fact it was engineered for a feature selection competition and has a sparse solution vector. In general our method significantly outperforms Gaussian mixture model clustering in the original feature space. A likely explanation is that (i) $f_\theta(X)|Y$ is more likely to be normal than $X|Y$ and (ii) it is easier to estimate in one dimensional space rather than in a high dimensional space.

2.4.1 Inaccurate Specification of $\mathbb{P}(Y)$

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

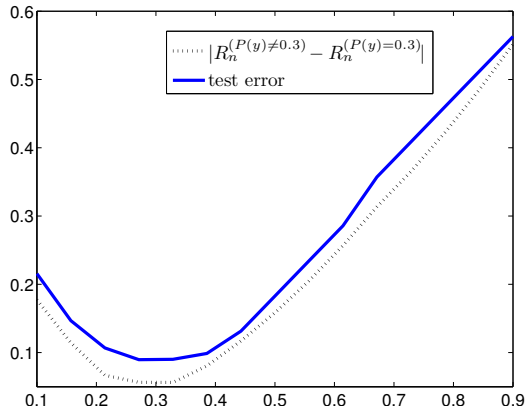


Figure 9: Performance of unsupervised classifier training on RCV1 data (top class vs. classes 2-5) for misspecified $\mathbb{P}(Y)$. The performance of the estimated classifier (in terms of training set empirical logloss R_n (5) and test error rate measured using held-out labels) decreases with the deviation between the assumed and true $\mathbb{P}(Y = 1)$ (true $\mathbb{P}(Y = 1) = 0.3$)). The classifier performance is very good when the assumed $\mathbb{P}(Y)$ is close to the truth and degrades gracefully when the assumed $\mathbb{P}(Y)$ is not too far from the truth.

$\mathbb{P}(Y)$.

Figure 9 displays the performance of the learned classifier for RCV1 data as a function of the assumed value of $\mathbb{P}(Y = 1)$ (correct value is $\mathbb{P}(Y = 1) = 0.3$). We conclude that knowledge of $\mathbb{P}(Y)$ is an important component in our framework but precise knowledge is not crucial. Small deviations of the assumed $\mathbb{P}(Y)$ from the true $\mathbb{P}(Y)$ result in a small degradation of logloss estimation quality and testing set error rate. Naturally, large deviation of the assumed $\mathbb{P}(Y)$ from the true $\mathbb{P}(Y)$ renders the framework ineffective.

2.4.2 Effect of Regularization and Dimensionality reduction.

In Figure 10 we examine the effect of regularization on the performance of the unsupervised classifier. In this experiment we use the L_1 regularization. Clearly, regularization helps in the supervised case. It appears that in the USL case weak regularization may improve performance but not as drastically as in the supervised case. Furthermore, the positive effect of L_1 regularization in the USL case appears to be weaker than L_2 regularization (compare the left and right panels of Figure 10). One

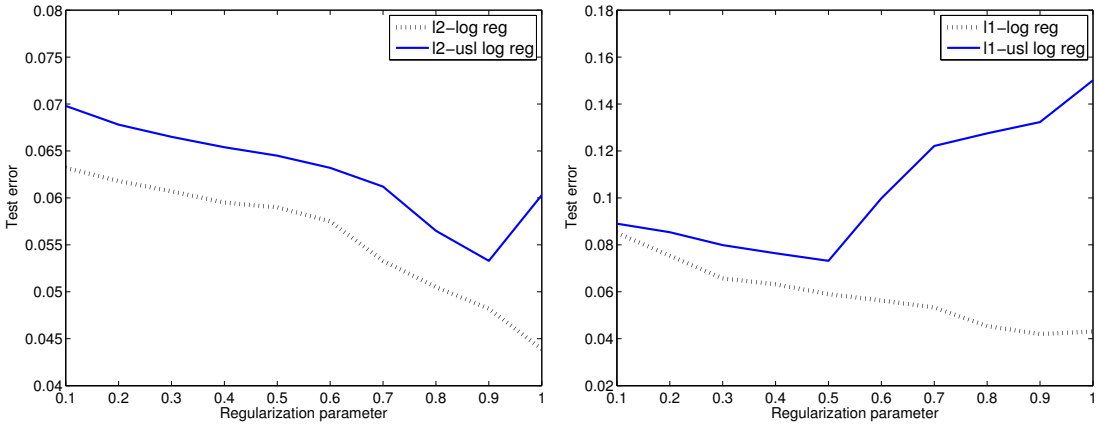


Figure 10: Test set error rate versus regularization parameter (L_2 on the left panel and L_1 on the right panel) for supervised and unsupervised logistic regression on RCV1 data set.

possible reason is that the sparsity promoting nature of L_1 conflicts with the CLT assumption.

In Figure 11 we examine the effect of reducing the data dimensionality via PCA prior to training the unsupervised classifier. Specifically, the 256 dimensions USPS image dataset was embedded in an increasingly lower dimensional space via PCA. For the original dimensionality of 256 or a slightly lower dimensionality the classification performance of the unsupervised classifier is comparable to the supervised. Once the dimensions are reduced to less than 150 a significant performance gap appears. This is consistent with our observation above that for lower dimensions the CLT approximation is less accurate. The supervised classifier also degrades in performance as less dimensions are used but not as fast as the unsupervised classifier.

2.5 Related Work

Semi-supervised approaches: Semisupervised learning is closely related to our work in that unsupervised classification may be viewed as a limiting case. One of the first attempts at studying the sample complexity of classification with unlabeled and labeled data was by [24]. They consider a setting when data is generated by mixture distributions and show that with infinite unlabeled data, the probability of error

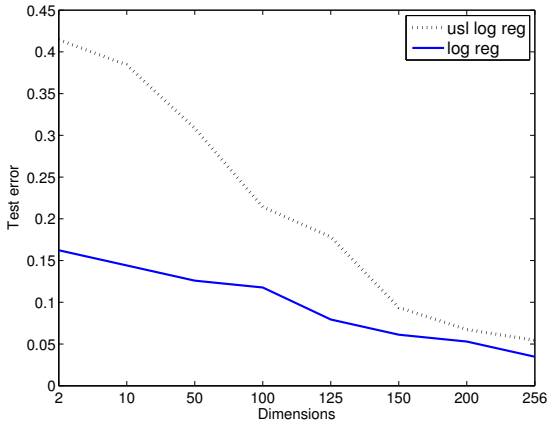


Figure 11: Test set error rate versus the amount of dimensions used (extracted via PCA) for supervised and unsupervised logistic regression on USPS data set. The original dimensionality was 256.

decays exponentially faster in the labeled data to the Bayes risk. They also analyze the case when there are only finite labeled and unlabeled data samples, with known class conditional densities but unknown mixing proportions [25]. A variant of the same scenario with known parametric forms for the class conditionals (specifically n-dimensional Gaussians) but unknown parameters and mixing proportions is also analyzed by [59]. Some of the more recent work in the area concentrated on analyzing semisupervised learning under the cluster assumption or the manifold assumption. We refer the reader to a recent survey by [131] for a discussion of recent approaches. However, none of the prior work consider mixture modeling in the projected 1-d space along with a CLT assumption which we exploit. In addition, assuming known mixing proportions, we propose a framework for training a classifier with no labeled samples, while approaches above still need labeled samples for classification.

Unsupervised approaches: The most recent related research approaches are by [81], [46], and [33]. The work by [81] aims to estimate the labels of an unlabeled testing set using known label proportions of several sets of unlabeled observations. The key difference between their approach and ours is that they require separate training sets from different sampling distributions with different and known label marginals

(one for each label). Our method assumes only a single dataset with a known label marginal but on the other hand assumed the CLT approximation. Furthermore, as noted previously (see comment after Proposition 5), our analysis is in fact valid when only the order of label proportions is known, rather than the absolute values.

A different attempt at solving this problem is provided by [46] which focuses on discriminative clustering. This approach attempts to estimate a conditional probabilistic model in an unsupervised way by maximizing mutual information between the empirical input distribution and the label distribution. A key difference is the focus on probabilistic classifiers and in particular logistic regression whereas our approach is based on empirical risk minimization which also includes SVM. Another key difference is that the work by [46] lacks consistency results which characterize when it works from a theoretical perspective. The approach by [33] focuses on estimating the error rate of a given stochastic classifier (not necessarily linear) without labels. It is similar in that it estimates the 0/1 risk rather than the margin based risk. However, it uses a different strategy and it replaces the CLT assumption with a symmetric noise assumption.

An important distinction between our work and the references above is that our work provides an estimate for the margin-based risk and therefore leads naturally to unsupervised versions of logistic regression and support vector machines. We also provide asymptotic analysis showing convergence of the resulting classifier to the optimal classifier (minimizer of (1)). Experimental results show that in practice the accuracy of the unsupervised classifier is on the same order (but slightly lower naturally) as its supervised analog.

2.6 Computing $M_{m,n}$ for Section 2.2.4

In this section, we provide derivations for computing

$$M_{m,n}\Big(p_{\eta_i}(z), p_{\eta_j}(z)\Big) = \int_{-\infty}^{\infty} \left(\frac{z-\mu_i}{\sigma_i}\right)^m \left(\frac{z-\mu_j}{\sigma_j}\right)^n \frac{p_{\eta_i}(z)p_{\eta_j}(z)}{p_\eta(z)}\, dz$$

from Section 2.2.4 using a power series expansion. We follow the transformation technique described in [9].

Without loss of generality we assume $\sigma_1 \leq \sigma_{-1}$ and consider the linear transformation

$$t = \epsilon \left(z - \bar{\mu}\right) / \bar{\sigma}$$

where

$$\epsilon = \begin{cases} 1 & \mu_1 \leq \mu_{-1} \\ -1 & \mu_1 > \mu_{-1} \end{cases}$$
$$\bar{\mu} = \frac{\mu_1 + \mu_{-1}}{2}$$
$$\bar{\sigma} = \sqrt{\sigma_1 \sigma_{-1}}.$$

Denoting $D = |\mu_{-1} - \mu_1|/2\bar{\sigma}$ and $r = \sigma_1/\sigma_{-1}$ the density function given by

$$p_\eta(z) = p_1 p_{\eta_1}(z) + p_{-1} p_{\eta_{-1}}(z) \tag{18}$$

is equivalent to

$$g(t) = p_1 g_1(t) + p_{-1} g_{-1}(t)$$

where

$$g_i(t) = \frac{1}{\sqrt{2\pi} r_i} \exp - \frac{(t - D_i)^2}{2 r_i} \qquad \text{for } i = 1, -1.$$

with $D_1 = -D$ and $D_{-1} = D$, $r_1 = r$ and $r_{-1} = 1/r$.

This transformation reduces normal mixtures with four parameters to an equivalent standard normal mixture with two parameters D, r. Hence the integral could be written as

$$M_{m,n}\Big(p_{\eta_i}(z), p_{\eta_j}(z)\Big) = \epsilon^{m+n} r_i^{-m/2} r_j^{-n/2} G_{m,n}(g_i, g_j)$$

where

$$G_{m,n}(g_i, g_j) = \int_{-\infty}^{\infty} (t - D - i)^m (t - D_j)^n \frac{g_i(t) g_j(t)}{g(t)}\, dt$$

Observe that

$$\frac{g_i(t) g_j(t)}{g(t)} = \frac{1}{\sqrt{2\pi} r_i r_j / r} \frac{h_i(t) h_j(t)}{h(t)}$$

where $h(t) = p_1 h_1(t) + p_{-1} h_{-1}(t)$ with $h_i(t) = \exp - \frac{(t - D_i)^2}{2 r_i}$ for $i = 1, -1$ and α_1, α_{-1} are the roots of the equation

$$(1 - r^2)t^2 + 2D(1 + r^2)t + D^2(1 - r^2) - 2r \log(p_1 / p_{-1} r) = 0.$$

The geometric series expansion for $G_{m,n}(g_i, g_j)$ [9] is given by

$$G_{m,n}(g_i, g_j) = \frac{1}{\sqrt{2\pi} r_i r_j / r} \sum_{N=0}^{\infty} \left(\int_{-\infty}^{\alpha_1} H_N(t) dt + \int_{\alpha_1}^{\alpha_{-1}} \bar{H}_N(t) dt + \int_{\alpha_{-1}}^{\infty} H_N(t) dt \right)$$

where

$$\begin{aligned}
H_N(t) &= (t - D_i)^m (t - D_j)^n \phi_N(t) \\
\bar{H}_N(t) &= (t - D_i)^m (t - D_j)^n \bar{\phi}_N(t) \\
\phi_N(t) &= \frac{1}{p_{-1} r} \left(- \frac{p_1}{p_{-1} r} \right)^N \frac{h_i(t) h_j(t)}{h_{-1}(t)} \left(\frac{h_1(t)}{h_{-1}(t)} \right)^N \\
\bar{\phi}_N(t) &= \frac{1}{p_1} \left(- \frac{p_{-1} r}{p_1} \right)^N \frac{h_i(t) h_j(t)}{h_1(t)} \left(\frac{h_{-1}(t)}{h_1(t)} \right)^N .
\end{aligned}$$

Since $\phi(t)$ and $\bar{\phi}(t)$ are constant multiples of normal densities, the computation of the integral corresponds to that of truncated non-central moments of the normal distribution. For example,

$$\begin{aligned}
M_{0,0}\Big(p_{\eta_1}(z), p_{\eta_{-1}}(z) \Big) &= G_{00}(g_1, g_{-1}) \\
&= \frac{1}{p_{-1} r} \sum_{N=0}^{\infty} (-1)^N r \sqrt{N(1 - r^2) + 1} \exp\Big(\frac{2D^2 r N(N = 1)}{N(1 - r^2) + 1} - \frac{N}{2r} \Big).
\end{aligned}$$

CHAPTER III

ESTIMATING CLASSIFICATION AND REGRESSION ERRORS WITHOUT LABELS

3.1 Introduction

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 $\mathbb{P}(X, Y)$ and a loss function $L(Y, \hat{Y})$, a predictor $f : \mathcal{X} \to \mathcal{Y}$ is characterized by an expected loss or risk function

$$R(f) = \mathsf{E}_{\mathbb{P}(X,Y)}\{\mathcal{L}(Y, f(X))\}. \tag{19}$$

For example, in classification we may have $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{1, \ldots, l\}$, and $\mathcal{L}(Y, \hat{Y}) = I(Y \neq \hat{Y})$ 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

$$R(f) = P(f \text{ predicts the wrong class}). \tag{20}$$

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

$$R(f) = \mathsf{E}_{\mathbb{P}(X,Y)}(Y - f(X))^2. \tag{21}$$

We consider the case where we are provided with k predictors $f_i : \mathcal{X} \to \mathcal{Y}$, $i = 1, \ldots, k$ ($k \geq 1$) whose risks are unknown. The main task we are faced with is estimating the risks $R(f_1), \ldots, R(f_k)$ 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 $X^{(1)}, \ldots, X^{(n)} \overset{\text{iid}}{\sim} \mathbb{P}(X)$.

A secondary task that we consider is obtaining effective schemes for combining k predictors $f_1, \ldots, f_k$ 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, as we show in this chapter, if the marginal $\mathbb{P}(Y)$ is known it is possible in some cases to obtain a consistent estimator for the risks using only unlabeled data i.e.,

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

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 $R(f_1), \ldots, R(f_k)$ (risks). We also demonstrate that the proposed estimation technique works well in practice on both synthetic and real world data.

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

There are several reasons that motivate our approach of using exclusively unlabeled data to estimate the risks. Labeled data 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 $f_1, \ldots, f_k$ 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 $R(f_1), \ldots, R(f_k)$ using a separate unlabeled dataset.

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.

The collaborative nature of this diagnosis is especially useful for multiple predictors as the predictor ensemble $\{f_1, \ldots, f_k\}$ 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 3.3 we discuss extensions to the general framework and in Section 3.4-3.5 we discuss the theory underlying our estimation process. In Section 3.6 we discuss practical optimization algorithms. Section 4.6 contains an experimental study. We conclude with a discussion in Section 3.8.

3.2 Unsupervised Risk Estimation Framework

We adopt the framework presented in Section 3.1 with the added requirement that the predictors $f_1, \ldots, f_k$ 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 $\mathbb{Q}_i$ and predicts Y' with probability $\mathbb{Q}_i(Y'|X)$.

As mentioned previously our goal is to estimate the risk associated with classification or regression models $f_1, \ldots, f_k$ based on unlabeled data $X^{(1)}, \ldots, X^{(n)} \overset{\text{iid}}{\sim} \mathbb{P}(X)$. The testing marginal and conditional distributions $\mathbb{P}(X), \mathbb{P}(Y|X)$ 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 $R(f_1), \ldots, R(f_k)$ i.e. $R(f_j) = g_j(\theta)$ for some function $g_j : \Theta \to \mathbb{R}$, $j = 1, \ldots, k$. The parameter vector θ is estimated from data by connecting it to the probabilities

$$\mathbb{P}_j(Y'|Y) \overset{\text{def}}{=} p(f_j \text{ predicts } Y' | \text{ true label is } Y).$$

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

$$\mathbb{P}_\theta(\hat{Y}) = \int \mathbb{P}_\theta(\hat{Y}|Y)\mathbb{P}(Y)\, dy$$

. The precise equations are:

$$\hat{R}(f_j\,;\hat{Y}^{(1)},\ldots,\hat{Y}^{(n)}) = g_j(\hat{\theta}^{\mathrm{mle}}(\hat{Y}^{(1)},\ldots,\hat{Y}^{(n)})) \quad \text{where} \tag{22}$$

$$\hat{Y}^{(i)} \stackrel{\mathrm{def}}{=} (\hat{Y}_1^{(i)},\ldots,\hat{Y}_k^{(i)})$$

$$\hat{Y}_j^{(i)} \stackrel{\mathrm{def}}{=} f_j(X^{(i)})$$

$$\hat{\theta}^{\mathrm{mle}}(\hat{Y}^{(1)},\ldots,\hat{Y}^{(n)}) = \arg\max \ell(\theta\,;\hat{Y}^{(1)},\ldots,\hat{Y}^{(n)}) \tag{23}$$

$$\ell(\theta\,;\hat{Y}^{(1)},\ldots,\hat{Y}^{(n)}) = \sum_{i=1}^{n} \log \mathbb{P}_\theta(\hat{Y}_1^{(i)},\ldots,\hat{Y}_k^{(i)}) \tag{24}$$

$$= \sum_{i=1}^{n} \log \int_{\mathcal{Y}} \mathbb{P}_\theta(\hat{Y}_1^{(i)},\ldots,\hat{Y}_k^{(i)}|Y^{(i)})\mathbb{P}(Y^{(i)})\, d\mu(Y^{(i)}).$$

The integral in (24) 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 μ, depending on the topology of $\mathcal{Y}$. Note that (24) and its maximizer are computable without any labeled data. All that is required are the classifiers (as black boxes), unlabeled data $X^{(1)},\ldots,X^{(n)}$, and the marginal label distribution $\mathbb{P}(Y)$.

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

$$\hat{Y}^{\mathrm{new}} = \arg\max_{y\in\mathcal{Y}} \mathbb{P}_{\hat{\theta}^{\mathrm{mle}}}(Y \mid f_1(X^{\mathrm{new}}),\ldots,f_k(X^{\mathrm{new}}))$$

$$= \arg\max_{Y\in\mathcal{Y}} \mathbb{P}(Y) \prod_{j=1}^{k} \mathbb{P}_{\hat{\theta}_j^{\mathrm{mle}}}(f_j(X^{\mathrm{new}}) \mid Y). \tag{25}$$

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 $\hat{\theta}^{\mathrm{mle}}$ and $\hat{R}$ (consistency, asymptotic variance). Second, how can we efficiently solve the maximization problem (23). And

third, how does the framework work in practice. We address these three questions in Sections 3.4-3.5, 3.6, 4.6 respectively, We devote the rest of the current section to examine some important special cases of (23)-(24) and consider some generalizations in the next section.

3.2.1 Non-Collaborative Estimation of the Risks

In the non-collaborative case we estimate the risk of each one of the predictors $f_1, \ldots, f_k$ 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 (22)-(24) 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

$$\hat{R}(f\,; \hat{Y}^{(1)}, \ldots, \hat{Y}^{(n)}) = g(\hat{\theta}^{\mathrm{mle}}(\hat{Y}^{(1)}, \ldots, \hat{Y}^{(n)})) \tag{26}$$

$$\hat{\theta}^{\mathrm{mle}}(\hat{Y}^{(1)}, \ldots, \hat{Y}^{(n)}) = \arg\max_{\theta} \sum_{i=1}^{n} \log \int_{\mathcal{Y}} \mathbb{P}_\theta(\hat{Y}^{(i)}|Y^{(i)}) \mathbb{P}(Y^{(i)})\, d\mu(Y^{(i)}) \tag{27}$$

where $\hat{Y}^{(i)} = f(X^{(i)})$.

We consider below several important special cases.

3.2.1.1 Classification

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

$$\mathbb{P}_\theta(\hat{Y}|Y) = \theta_{\hat{Y},Y}. \tag{28}$$

where θ 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 θ is

$$R(f) = 1 - \sum_{y \in \mathcal{Y}} \theta_{YY}\, \mathbb{P}(Y). \tag{29}$$

Equations (28)-(29) may be simplified by assuming a symmetric error distribution [28]

$$\mathbb{P}_\theta(\hat{Y}|Y) = \theta^{I(\hat{Y}=Y)} \left(\frac{1-\theta}{l-1}\right)^{I(\hat{Y}\neq Y)} \tag{30}$$

$$R(f) = 1 - \theta \tag{31}$$

where I is the indicator function and $\theta \in [0,1]$ is a scalar corresponding to the classifier accuracy. Estimating θ by maximizing (27), with (28) or (30) substituting $\mathbb{P}_\theta$ completes the risk estimation task.

In the simple binary case $l = 2, \mathcal{Y} = \{1,2\}$ with the symmetric noise model (30) the loglikelihood

$$\ell(\theta) = \sum_{i=1}^{n} \log \sum_{Y^{(i)}=1}^{2} \theta^{I(\hat{Y}^{(i)}=Y^{(i)})}(1-\theta)^{I(\hat{Y}^{(i)}\neq Y^{(i)})}\mathbb{P}(Y^{(i)}). \tag{32}$$

may be shown to have the following closed form maximizer

$$\hat{\theta}^{\text{mle}} = \frac{\mathbb{P}(Y=1) - m/n}{2\mathbb{P}(Y=1) - 1}. \tag{33}$$

where $m \stackrel{\text{def}}{=} |\{i \in \{1,\ldots,n\} : \hat{Y}^{(i)} = 2\}|$. The estimator (33) 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 (30) does not hold, using (33) to estimate the classification risk may be misleading. For example, in some cases (33) may be negative. In these cases, using the more general model (28) instead of (30) should provide more accurate results. We discuss this further from theoretical and experimental perspectives in Sections 3.4-3.5, and 4.6 respectively.

3.2.1.2 Regression

Assuming a regression equation

$$Y = aX + \epsilon, \qquad \epsilon \sim N(0, \tau^2)$$

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 $\mathbb{P}_\theta(\hat{Y}|Y)$ and the relationship between the risk and the parameter $R(f) = g(\theta)$ are

$$\mathbb{P}_\theta(\hat{Y}|Y) = (2\pi\theta^2\tau^2)^{-1/2} \exp\left(-\frac{(\hat{Y} - \theta Y)^2}{2\theta^2\tau^2}\right) \tag{34}$$

$$R(f|Y) = \mathsf{bias}^{\,2}(f) + \mathsf{Var}\,(f) = (1-\theta)^2 y^2 + \theta^2\tau^2 \tag{35}$$

$$R(f) = \theta^2\tau^2 + (1-\theta)^2 \mathsf{E}_{\mathbb{P}(Y)}(Y^2). \tag{36}$$

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 $\mathbb{P}(Y) = N(\mu_Y, \sigma_Y^2)$ (as is often done in regression analysis) we have $\mathbb{P}_\theta(\hat{Y}^{(i)}) =$

$$= \int_{\mathbb{R}} \mathbb{P}_\theta(\hat{Y}^{(i)}|Y)\mathbb{P}(Y)dy = (2\pi\theta^2\tau^2 2\pi\sigma_Y^2)^{-1/2} \int_{\mathbb{R}} \exp\left(-\frac{(\hat{Y} - \theta Y)^2}{2\theta^2\tau^2} - \frac{(Y - \mu_Y)^2}{2\sigma_Y^2}\right) dy \tag{37}$$

$$= \frac{1}{\theta\sqrt{2\pi(\tau^2 + \sigma_Y^2)}} \exp\left(\frac{(\hat{Y}^{(i)})^2}{2\theta^2\tau^2}\left(\frac{\sigma_Y^2}{\sigma_Y^2 + \tau^2} - 1\right) + \frac{\mu_Y^2}{2\sigma_Y^2}\left(\frac{\tau^2}{\sigma_Y^2 + \tau^2} - 1\right) + \frac{\hat{Y}^{(i)}\mu_Y}{\theta\,(\tau^2 + \sigma_Y^2)}\right) \tag{38}$$

where we used the following lemma in the last equation.

Lemma 1 (e.g., [79])**.**

$$\int_{-\infty}^{\infty} A\, e^{-Bx^2 + Cx + D}\, dx = A\sqrt{\frac{\pi}{B}}\, \exp\left(C^2/4B + D\right) \tag{39}$$

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

In this case the loglikelihood simplifies to

$$\ell(\theta) = -n\log\left(\theta\sqrt{2\pi(\tau^2 + \sigma_Y^2)}\right) - \left(\frac{\sum_{i=1}^n (\hat{Y}^{(i)})^2}{2(\tau^2 + \sigma_Y^2)}\right)\frac{1}{\theta^2} + \left(\frac{\mu_Y \sum_{i=1}^n \hat{Y}^{(i)}}{\tau^2 + \sigma_Y^2}\right)\frac{1}{\theta} - n\frac{\mu_Y^2}{2(\sigma_Y^2 + \tau^2)} \tag{40}$$

which can be shown to have the following closed form maximizer

$$\hat{\theta}^{\text{mle}} = -\frac{\mu_Y \sum_{i=1}^n \hat{Y}^{(i)}}{2n(\tau^2+\sigma_Y^2)} \pm \sqrt{\frac{\left(\mu_Y \sum_{i=1}^n \hat{Y}^{(i)}\right)^2}{4n^2(\tau^2+\sigma_Y^2)^2} + \frac{\sum_{i=1}^n (\hat{Y}^{(i)})^2}{n(\tau^2+\sigma_Y^2)}} \tag{41}$$

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

The univariate regression case described above may be extended to multiple explanatory variables i.e., $Y = aX + \epsilon$ where Y, X, ϵ are vectors and A is a matrix. This is an interesting extension which falls beyond the scope of the current chapter.

3.2.1.3 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 $R(f) = \mathsf{E}\,\|Y - \hat{Y}\|^2$. In this case the distribution $\mathbb{P}_\theta(\hat{Y}|Y)$ and the relationship between the risk and the parameter $R(f) = g(\theta)$ are

$$\mathbb{P}_\theta(\hat{Y}|Y) = (2\pi\theta^2)^{-1/2} \exp\left(-\frac{(\hat{Y}-Y)^2}{2\theta^2}\right) \tag{42}$$

$$R(f|Y) = \theta^2 \tag{43}$$

$$R(f) = \theta^2 \mathsf{E}_{\,\mathbb{P}(Y)}(Y). \tag{44}$$

The loglikelihood and other details in this case are straightforward variations on the linear regression case described above. We therefore concentrate in this chapter on the classification and linear regression cases.

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 $\hat{\theta}_1^{\text{mle}}, \ldots, \hat{\theta}_k^{\text{mle}}$ 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 $R(f_1), \ldots, R(f_k)$ 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.

3.2.2 Collaborative Estimation of the Risks: Conditionally Independent Predictors

We have previously seen how to estimate the risks of k predictors by separately applying (22) to each predictor. If the predictors are known to be conditionally independent given the true label i.e. $\mathbb{P}_\theta(\hat{Y}_1, \ldots, \hat{Y}_k|Y) = \prod_j \mathbb{P}_{\theta_j}(\hat{Y}_j|Y)$ the loglikelihood (24) simplifies to

$$\ell(\theta) = \sum_{i=1}^{n} \log \int_{\mathcal{Y}} \prod_{j=1}^{k} \mathbb{P}_{\theta_j}(\hat{Y}_j^{(i)}|Y^{(i)}) \mathbb{P}(Y^{(i)})\, d\mu(Y^{(i)}), \quad \text{where} \quad \hat{Y}_j^{(i)} = f_j(X^{(i)}) \tag{45}$$

and $\mathbb{P}_{\theta_j}$ above is (28) or (30) for classification and (34) for regression. Maximizing the loglikelihood (45) jointly over $\theta_1, \ldots, \theta_k$ results in estimators $\hat{R}(f_1), \ldots, \hat{R}(f_k)$ that converge to the true value faster than the non-collaborative MLE (27) (more on this in Section 4.6). Equation (45) 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 $T_1, \ldots, T_k$ 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 $T_1, \ldots, T_k$ in an independent manner.

Figure 12 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 θ^{true}. Figure 13 displays a similar figure for $k = 1$ in the

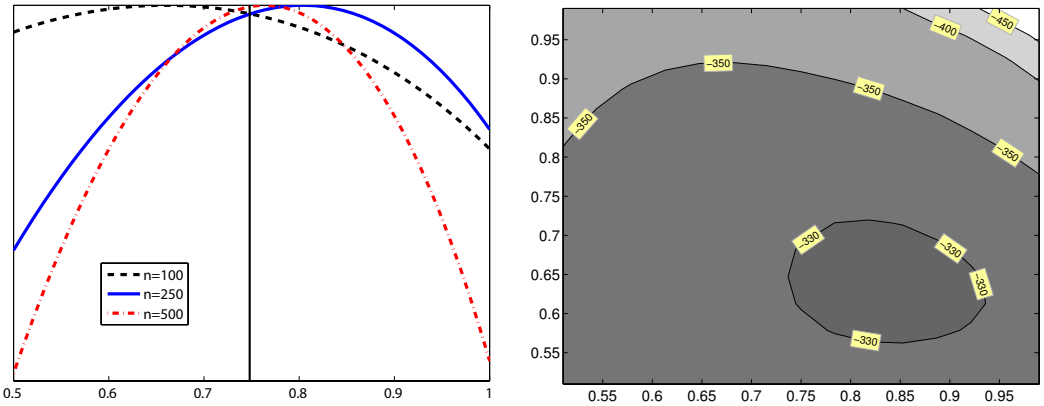


Figure 12: A plot of the loglikelihood functions $\ell(\theta)$ in the case of classification for $k = 1$ (left, $\theta^{\text{true}} = 0.75$) and $k = 2$ (right, $\theta^{\text{true}} = (0.8, 0.6)^\top$). The loglikelihood was constructed based on random samples of unlabeled data with sizes $n = 100, 250, 500$ (left) and $n = 250$ (right) and $\mathbb{P}(Y = 1) = 0.75$. 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 θ^{true}.

case of regression.

In the case of regression (45) 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 (45) simplifies to

$$\mathbb{P}_\theta(\hat{Y}_1^{(i)}, \ldots, \hat{Y}_k^{(i)}) =$$

$$\int_{-\infty}^{\infty} \left(\prod_{j=1}^{k} \frac{1}{\theta_j \tau \sqrt{2\pi}} e^{-\left(\hat{Y}_j^{(i)} - \theta_j Y^{(i)}\right)^2 \Big/ 2\theta_j^2 \tau^2} \right) \frac{1}{\sigma_y \sqrt{2\pi}} e^{-\left(Y^{(i)} - \mu_Y\right)^2 \Big/ 2\sigma_y^2} \, dY^{(i)}$$

$$= \frac{1}{\tau^k (\sqrt{2\pi})^{k+1} \sigma_Y \prod_{j=1}^{k} \theta_j} \int_{-\infty}^{\infty} \exp\left[-\frac{1}{2} \left(\left(\frac{Y^{(i)} - \mu_Y}{\sigma_y} \right)^2 + \sum_{j=1}^{k} \left(\frac{Y^{(i)}}{\tau} - \frac{\hat{Y}_j^{(i)}}{\tau \theta_j} \right)^2 \right) \right] dY^{(i)}$$

$$= \frac{\int_{-\infty}^{\infty} \exp\left(-\frac{1}{2} \left(\frac{1}{\sigma_Y^2} + \frac{k}{\tau^2} \right) (Y^{(i)})^2 + \left(\frac{\mu_Y}{\sigma_Y^2} + \sum_{j=1}^{k} \frac{\hat{Y}_j^{(i)}}{\tau^2 \theta_j} \right) Y^{(i)} - \frac{1}{2} \left(\frac{\mu_Y^2}{\sigma_Y^2} + \sum_{j=1}^{k} \frac{(\hat{Y}_j^{(i)})^2}{\tau^2 \theta_j^2} \right) \right)}{\tau^k (\sqrt{2\pi})^{k+1} \sigma_Y \prod_{j=1}^{k} \theta_j}$$

$$= \frac{\sqrt{\pi} \left[\frac{1}{2} \left(\frac{1}{\sigma_Y^2} + \frac{k}{\tau^2} \right) \right]^{-1/2}}{\tau^k (\sqrt{2\pi})^{k+1} \sigma_Y \prod_{j=1}^{k} \theta_j} \exp\left(\frac{\left(\frac{\mu_Y}{\sigma_Y^2} + \sum_{j=1}^{k} \frac{\hat{Y}_j^{(i)}}{\tau^2 \theta_j} \right)^2}{2 \left(\frac{1}{\sigma_Y^2} + \frac{k}{\tau^2} \right)} - \sum_{j=1}^{k} \frac{(\hat{Y}_j^{(i)})^2}{2\tau^2 \theta_j^2} - \frac{\mu_Y^2}{2\sigma_Y^2} \right) \tag{46}$$

where the last equation was obtained using Lemma 1 concerning Gaussian integrals. Note that this equation does not have a closed form maximizer requiring the use of

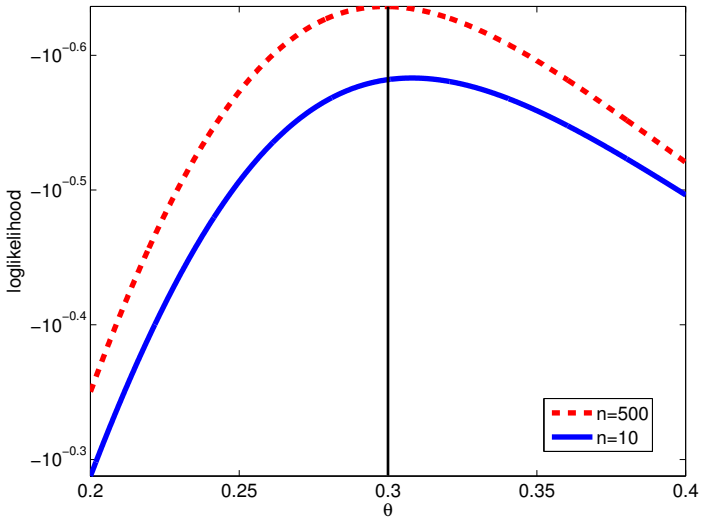


Figure 13: A plot of the loglikelihood function $\ell(\theta)$ in the case of regression for $k = 1$ with $\theta^{\text{true}} = 0.3$, $\tau = 1$, $\mu_Y = 0$ and $\sigma_Y = 0.2$. As additional unlabeled examples are added the loglikelihood curve become steeper and their maximizers get closer to the true parameter θ^{true} resulting in a more accurate risk estimate.

iterative computational techniques.

3.2.3 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 (45) 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 [17]. For example, generalizing our conditional independence assumption to second-order interaction log-linear models we have

$$\log p(\hat{Y}_1, \ldots, \hat{Y}_k | Y) = \alpha_y + \sum_{i=1}^{l} \beta_{i,\hat{Y}_i,Y} + \sum_{i<j} \gamma_{i,j,\hat{Y}_i,\hat{Y}_j,Y} \tag{47}$$

where the following ANOVA-type parameter constraints are needed [17]

$$0 = \sum_{\hat{Y}_i} \beta_{i,\hat{Y}_i,Y} \quad \forall i, Y \tag{48}$$

$$0 = \sum_{\hat{Y}_i} \gamma_{i,j,\hat{Y}_i,\hat{Y}_j,Y} = \sum_{\hat{Y}_j} \gamma_{i,j,\hat{Y}_i,\hat{Y}_j,Y} \quad \forall i, j, Y.$$

The β parameters in (47) correspond to the order-1 interaction between the variables $\hat{Y}_1, \ldots, \hat{Y}_k$, conditioned on Y. They correspond to the θ_i in the independent formulation (28)-(30). The γ 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 (28)-(30).

In the case of classification, the number of degrees of freedom or free unconstrained parameters in (47) depends on whether the number of classes is 2 or more and what additional assumptions exist on β and γ. 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 3.2.2 work as before with a slight modification of replacing (28)-(30) with variations based on (47) and enforcing the constraints (48).

Equation (47) captures two-way interactions but cannot model higher order interactions. However, three-way and higher order interaction models are straightforward generalizations of (47) culminating in the full loglinear model which does not make any assumption on the statistical dependency of the noise operators $T_1, \ldots, T_k$. 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 $R(f_1), \ldots, R(f_k)$.

In our experiments on real world data (see Section 4.6), it is often the case that maximizing the loglikelihood under the conditionally independent assumption (45) provides adequate accuracy and there is no need for the more general (47)-(48).

Nevertheless, we include here the case of loglinear models as it may be necessary in some situations.

3.3 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 $X^{(1)}, \ldots, X^{(n)}$ each predictor may provide output on an arbitrary subset of the data points $\{f_j(X^{(i)}) : i \in S_j\}$, where $S_j \subset \{1, \ldots, n\}$, $j = 1, \ldots, k$.

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 $R(f_1), \ldots, R(f_k)$. 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 β_{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

$$\begin{aligned}\hat{\theta}_n^{\text{mle}} &= \arg\max_{\theta} \ell(\theta) \\ \ell(\theta) &= \sum_{i=1}^{n} \log \sum_{r:\beta_{ri}=0} \int_{\mathcal{Y}} \mathbb{P}_\theta(\hat{Y}_1^{(i)}, \ldots, \hat{Y}_k^{(i)})\, d\mu(\hat{Y}_r^{(i)}) \\ &= \sum_{i=1}^{n} \log \sum_{r:\beta_{ri}=0} \iint_{\mathcal{Y}^2} \mathbb{P}_\theta(\hat{Y}_1^{(i)}, \ldots, \hat{Y}_k^{(i)}|Y^{(i)})\mathbb{P}(Y^{(i)})\, d\mu(\hat{Y}_r^{(i)}) d\mu(Y^{(i)})\end{aligned} \tag{49}$$

where $\mathbb{P}_\theta$ 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

$$\ell(\theta) = \sum_{i=1}^{n} \log \int_{\mathcal{Y}} \prod_{j=1}^{k} \mathbb{P}_{\theta_j}(\hat{Y}_j^{(i)}|Y^{(i)})\mathbb{P}(Y^{(i)})\, d\mu(Y^{(i)}) + \sum_{i=n+1}^{m} \log \prod_{j=1}^{k} \mathbb{P}_{\theta_j}(\hat{Y}_j^{(i)}|Y^{(i)})\mathbb{P}(Y^{(i)}). \tag{50}$$

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.

3.4 Consistency of $\hat{\theta}_n^{mle}$ 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 (23) and the risk estimator $\hat{R}(f_j) = g_j(\hat{\theta}^{\text{mle}})$ defined in (22). The analysis is conducted under the assumption that the vectors of observed predictors outputs $\hat{Y}^{(i)} = (\hat{Y}_1^{(i)}, \ldots, \hat{Y}_k^{(i)})$ are iid samples from the distribution

$$\mathbb{P}_\theta(\hat{Y}) = \mathbb{P}_\theta(\hat{Y}_1, \ldots, \hat{Y}_k) = \int_{\mathcal{Y}} \mathbb{P}_\theta(\hat{Y}_1, \ldots, \hat{Y}_k|Y)\mathbb{P}(Y)\, d\mu(y).$$

We start by investigating whether estimator $\hat{\theta}^{\text{mle}}$ in (23) converges to the true parameter value. More formally, strong consistency of the estimator $\hat{\theta}_n^{\text{mle}} = \hat{\theta}(\hat{Y}^{(1)}, \ldots, \hat{Y}^{(n)})$, $\hat{Y}^{(1)}, \ldots, \hat{Y}^{(n)} \overset{\text{iid}}{\sim} \mathbb{P}_{\theta_0}$ is defined as strong convergence of the estimator to θ_0 as $n \to \infty$ [43]

$$\lim_{n\to\infty} \hat{\theta}_n^{\text{mle}}(\hat{Y}^{(1)}, \ldots, \hat{Y}^{(n)}) = \theta_0 \text{ with probability 1.} \tag{51}$$

In other words as the number of samples n grows, the estimator will surely converge to the true parameter θ_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 11 (e.g., [43])**.** *Let $\hat{Y}^{(1)}, \ldots, \hat{Y}^{(n)} \overset{\text{iid}}{\sim} \mathbb{P}_{\theta_0}$, $\theta_0 \in \Theta$. If the following conditions hold*

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

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

Note that $\mathbb{P}_\theta(\hat{Y})$ in the proposition above corresponds to $\int_{\mathcal{Y}} \mathbb{P}_\theta(\hat{Y}|Y)\mathbb{P}(Y)\, d\mu(y)$

in our framework. That is the MLE operates on the observed data or predictor output $\hat{Y}^{(1)}, \ldots, \hat{Y}^{(n)}$ that is sampled iid from the distribution $\mathbb{P}_{\theta_0}(\hat{Y}) = \int_y \mathbb{P}_{\theta_0}(\hat{Y}|Y)\mathbb{P}(Y)\, d\mu(y)$.

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 $\int_y \mathbb{P}_\theta(\hat{Y}|Y)\mathbb{P}(Y)\, d\mu(y)$ is continuous in θ 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 θ') whose expectation converges to the KL divergence which in turn is never ∞ for Gaussian distributions or its derivatives. The fourth condition of measurability follows as $\mathbb{P}_\theta$ 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 $\mathbb{P}_\theta(\hat{Y})$ and $\mathbb{P}_{\theta_0}(\hat{Y})$ 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 (30). The non-symmetric case (28) is more complicated and is treated afterwards. We conclude the consistency discussion with an examination of the regression case.

3.4.1 Consistency of Classification Risk Estimation

Proposition 12. *Let $f_1, \ldots, f_k$ be classifiers $f_i : \mathcal{X} \to \mathcal{Y}$, $|\mathcal{Y}| = l$, with conditionally independent noise processes described by* (30)*. If the classifiers are weak learners i.e., $1/l < 1 - err(f_i) < 1$ and $\mathbb{P}(Y)$ is not uniform the unsupervised collaborative diagnosis model is identifiable.*

Corollary 1. *Let $f_1, \ldots, f_k$ be classifiers $f_i : \mathcal{X} \to \mathcal{Y}$ with $|\mathcal{Y}| = l$ and noise processes described by* (30)*. If the classifiers are weak learners i.e., $1/l < 1 - err(f_i) < 1$, and $\mathbb{P}(Y)$ is not uniform the unsupervised non-collaborative diagnosis model is identifiable.*

Proof. Proving identifiability in the non-collaborative case proceeds by invoking Proposition 12 (whose proof is given below) with $k = 1$ separately for each classifier. The conditional independence assumption in Proposition 12 becomes redundant in this case of a single classifier, resulting in identifiability of $\mathbb{P}_{\theta_j}(\hat{Y}_j)$ for each $j = 1, \ldots, k$ □

Corollary 2. *Under the assumptions of Proposition 12 or Corollary 1 the unsupervised maximum likelihood estimator is consistent i.e.,*

$$P\left(\lim_{n\to\infty} \hat{\theta}_n^{mle}(\hat{Y}^{(1)}, \ldots, Y^{(n)}) = (\theta_1^{true}, \ldots, \theta_k^{true})\right) = 1.$$

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

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

Proof. Proposition 12 or Corollary 1 establishes identifiability, which in conjunction with Proposition 11 proves the corollary. □

Proof. (**for Proposition 12**) We prove identifiability by induction on k. In the base

case of $k = 1$, we have a set of l equations, corresponding to $i = 1, 2 \ldots l$,

$$\begin{aligned}
\mathbb{P}_\theta(\hat{Y}_1 = i) &= \mathbb{P}(Y = i)\theta_1 + \left(\sum_{j \neq i} \mathbb{P}(Y = j)\right) \frac{(1 - \theta_1)}{(l - 1)} \\
&= \mathbb{P}(Y = i)\theta_1 + (1 - \mathbb{P}(Y = i))\frac{(1 - \theta_1)}{(l - 1)} \\
&= \frac{\theta_1(l\mathbb{P}(Y = i) - 1) + 1 - \mathbb{P}(Y = i)}{(l - 1)}
\end{aligned}$$

from which we can see that if $\eta \neq \theta$ and $\mathbb{P}(Y = i) \neq 1/l$ then $\mathbb{P}_\theta(\hat{Y}_1) \neq \mathbb{P}_\eta(\hat{Y}_1)$. 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 θ_{k+1}, η_{k+1}.

In the case of k classifiers we have

$$\mathbb{P}_\theta(\hat{Y}_1, \ldots, \hat{Y}_k) = \sum_{i=1}^{l} \mathbb{P}_\theta(\hat{Y}_1, \ldots, \hat{Y}_k | Y = i)\mathbb{P}(Y = i) = \sum_{i=1}^{l} G(\mathcal{A}_i, \theta)$$

where

$$G(\mathcal{A}_i, \theta) \stackrel{\text{def}}{=} \mathbb{P}(Y = i) \prod_{j \in \mathcal{A}_i} \theta_j \cdot \prod_{j \notin \mathcal{A}_i} \frac{(1 - \theta_j)}{(l - 1)}.$$

$$\mathcal{A}_i \stackrel{\text{def}}{=} \{j \in \{1, 2..., k\} : \hat{Y}_j = i\}.$$

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

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 $\hat{Y}_{k+1} = 1, 2, \ldots, l$) to hold

for any $\hat{Y}_1, \ldots, \hat{Y}_k$ which corresponds to any partition $\mathcal{A}_1, \ldots, \mathcal{A}_l$

$$\begin{aligned}
\theta_{k+1} G(\mathcal{A}_1, \theta) + \frac{(1-\theta_{k+1})}{(l-1)} \sum_{i \neq 1} G(\mathcal{A}_i, \theta) &= \eta_{k+1} G(\mathcal{A}_1, \eta) + \frac{(1-\eta_{k+1})}{(l-1)} \sum_{i \neq 1} G(\mathcal{A}_i, \eta) \\
\theta_{k+1} G(\mathcal{A}_2, \theta) + \frac{(1-\theta_{k+1})}{(l-1)} \sum_{i \neq 2} G(\mathcal{A}_i, \theta) &= \eta_{k+1} G(\mathcal{A}_2, \eta) + \frac{(1-\eta_{k+1})}{(l-1)} \sum_{i \neq 2} G(\mathcal{A}_i, \eta) \\
&\vdots \\
\theta_{k+1} G(\mathcal{A}_l, \theta) + \frac{(1-\theta_{k+1})}{(l-1)} \sum_{i \neq l} G(\mathcal{A}_i, \theta) &= \eta_{k+1} G(\mathcal{A}_l, \eta) + \frac{(1-\eta_{k+1})}{(l-1)} \sum_{i \neq l} G(\mathcal{A}_i, \eta).
\end{aligned} \tag{52}$$

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 $\mathbb{P}_\theta(\hat{Y}_1, \ldots, \hat{Y}_k, \hat{Y}_{k+1})$ and $\mathbb{P}_\eta(\hat{Y}_1, \ldots, \hat{Y}_k, \hat{Y}_{k+1})$ to provide

$$\sum_{i=1}^{l} G(\mathcal{A}_i, \theta) = \sum_{i=1}^{l} G(\mathcal{A}_i, \eta) \tag{53}$$

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 $t \in \{1, \ldots, l\}$ which simplifies to

$$0 = (\theta_{k+1} - \eta_{k+1}) \left(l G(\mathcal{A}_t, \theta) - \sum_{i=1}^{l} G(\mathcal{A}_i, \theta) \right) \quad t = 1, \ldots, k. \tag{54}$$

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

$$l G(\mathcal{A}_t, \theta) = \sum_{i=1}^{l} G(\mathcal{A}_i, \theta). \tag{55}$$

We show that (55) cannot hold by examining separately the cases $\mathbb{P}(Y = t) > 1/l$ and $\mathbb{P}(Y = t) < 1/l$. Recall that there exists a t for which $\mathbb{P}(Y = t) \neq 1/l$ since the proposition requires that $\mathbb{P}(Y)$ is not uniform.

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

$$l\mathbb{P}(Y=t)\prod_{j=1}^{k}\theta_j = \sum_{i\neq t}\mathbb{P}(Y=i)\prod_{j=1}^{k}\frac{1-\theta_j}{l-1} + \mathbb{P}(Y=t)\prod_{j=1}^{k}\theta_j$$

$$(l-1)\mathbb{P}(Y=t)\prod_{j=1}^{k}\theta_j = (1-\mathbb{P}(Y=t))\prod_{j=1}^{k}\frac{1-\theta_j}{l-1}$$

$$\mathbb{P}(Y=t)\prod_{j=1}^{k}\theta_j = \frac{(1-\mathbb{P}(Y=t))}{(l-1)}\prod_{j=1}^{k}\frac{1-\theta_j}{l-1}$$

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

$$l\mathbb{P}(Y=t)\prod_{j=1}^{k}\frac{1-\theta_j}{l-1} = \sum_{i\neq s}\mathbb{P}(Y=i)\prod_{j=1}^{k}\frac{1-\theta_j}{l-1} + \mathbb{P}(Y=s)\prod_{j=1}^{k}\theta_j$$

$$(l\mathbb{P}(Y=t) - p(y\neq s))\prod_{j=1}^{k}\frac{1-\theta_j}{l-1} = \mathbb{P}(Y=s)\prod_{j=1}^{k}\theta_j$$

which cannot hold as the term on the left hand side is necessarily smaller than the term on the right hand side (if $\mathbb{P}(Y=t) < 1/l$ 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 $k \geq 1$. □

The conditions asserted above that $\mathbb{P}(Y) \neq 1/l$ and $1/l < 1 - \mathrm{err}(f_i) < 1$ 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 θ in (28) is a 2×2 matrix with two degrees of freedom as each row sums to one. In particular we have $\theta_{11} = \mathbb{P}_\theta(\hat{Y}=1|Y=1)$, $\theta_{12} = \mathbb{P}_\theta(\hat{Y}=1|Y=2)$, $\theta_{21} = \mathbb{P}_\theta(\hat{Y}=2|Y=1)$, $\theta_{22} = \mathbb{P}_\theta(\hat{Y}=2|Y=2)$ with the

overall risk $R(f) = 1-\theta_{11}\mathbb{P}(Y=1)-\theta_{22}\mathbb{P}(Y=2)$. Unfortunately, the matrix θ is not identifiable in this case and neither is the scalar parameter $\theta_{11}\mathbb{P}(Y=1)+\theta_{22}\mathbb{P}(Y=2)$ that can be used to characterize the risk.

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

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

Proof. For two different parameterizations θ, η we have

$$\mathbb{P}_\theta(\hat{Y}=1) = \mathbb{P}(Y=1)\theta_{11} + (1-\mathbb{P}(Y=1))(1-\theta_{22}) \tag{56}$$

$$\mathbb{P}_\theta(\hat{Y}=2) = \mathbb{P}(Y=1)(1-\theta_{11}) + (1-\mathbb{P}(Y=1))\theta_{22} \tag{57}$$

and

$$\mathbb{P}_\eta(\hat{Y}=1) = \mathbb{P}(Y=1)\eta_{11} + (1-\mathbb{P}(Y=1))(1-\eta_{22}) \tag{58}$$

$$\mathbb{P}_\eta(\hat{Y}=2) = \mathbb{P}(Y=1)(1-\eta_{11}) + (1-\mathbb{P}(Y=1))\eta_{22}. \tag{59}$$

Equating the two Equations (56) and (58) we have

$$\mathbb{P}(Y=1)(\theta_{11}+\theta_{22}) + 1 - \mathbb{P}(Y=1) - \theta_{22} = \mathbb{P}(Y=1)(\eta_{11}+\eta_{22}) + 1 - \mathbb{P}(Y=1) - \eta_{22}$$

$$\mathbb{P}(Y=1)\theta_{11} - (1-\mathbb{P}(Y=1))\theta_{22} = \mathbb{P}(Y=1)\eta_{11} - (1-\mathbb{P}(Y=1))\eta_{22}$$

$$\mathbb{P}(Y=1)\theta_{11} - \mathbb{P}(Y=2)\theta_{22} = \mathbb{P}(Y=1)\eta_{11} - \mathbb{P}(Y=2)\eta_{22}$$

Similarly, equating Equation (57) and Equation (59) also results in $\mathbb{P}(Y=1)\theta_{11} - \mathbb{P}(Y=2)\theta_{22} = \mathbb{P}(Y=1)\eta_{11} - \mathbb{P}(Y=2)\eta_{22}$. As a result, we have

$$\mathbb{P}_\theta \equiv \mathbb{P}_\eta \quad \Rightarrow \quad \mathbb{P}(Y=1)\theta_{11} - \mathbb{P}(Y=2)\theta_{22} = \mathbb{P}(Y=1)\eta_{11} - \mathbb{P}(Y=2)\eta_{22}.$$

□

The above lemma indicates that we can use the maximum likelihood method to obtain a consistent estimator for the parameter $\theta_{11}\mathbb{P}(Y=1)-\theta_{22}\mathbb{P}(Y=2)$. Unfortunately the parameter $\theta_{11}\mathbb{P}(Y=1)-\theta_{22}\mathbb{P}(Y=2)$ 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 $\mathbb{P}_1(Y)$ and $\mathbb{P}_2(Y)$.

Proposition 13. *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.*

$$X^{(1)},\ldots,X^{(n)} \overset{\text{iid}}{\sim} \mathbb{P}_1(x)=\sum_Y \mathbb{P}(X|Y)\mathbb{P}_1(Y)$$
$$X'^{(1)},\ldots,X'^{(m)} \overset{\text{iid}}{\sim} \mathbb{P}_2(x)=\sum_Y \mathbb{P}(X|Y)\mathbb{P}_2(Y)$$

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 $\hat{Y}^{(1)},\ldots,\hat{Y}^{(n)}$, $\hat{Y}'^{(1)},\ldots,\hat{Y}'^{(m)}$ where $\hat{Y}^{(i)} \overset{\text{iid}}{\sim} \sum_y \mathbb{P}_\theta(\hat{Y}|Y)\mathbb{P}_1(Y)$ and $\hat{Y}'^{(i)} \overset{\text{iid}}{\sim} \sum_y \mathbb{P}_\theta(\hat{Y}|Y)\mathbb{P}_2(Y)$. In particular, note that the marginal distributions $\mathbb{P}_1(Y)$ and $\mathbb{P}_2(Y)$ are different but the parameter matrix θ is the same in both cases as we operate the same classifier on samples from the same class conditional distribution $\mathbb{P}(X|Y)$.

Based on Lemma 2 we construct a consistent estimator for $\mathbb{P}_1(Y=1)\theta_{11}-\mathbb{P}_1(Y=2)\theta_{22}$ by maximizing the likelihood of $\hat{Y}^{(1)},\ldots,\hat{Y}^{(n)}$. Similarly, we construct a consistent estimator for $\mathbb{P}_2(Y=1)\theta_{11}-\mathbb{P}_2(Y=2)\theta_{22}$ by maximizing the likelihood of $\hat{Y}'^{(1)},\ldots,\hat{Y}'^{(m)}$. Note that $\mathbb{P}_1(Y=1)\theta_{11}-\mathbb{P}_1(Y=2)\theta_{22}$ and $\mathbb{P}_2(Y=1)\theta_{11}-\mathbb{P}_2(Y=2)\theta_{22}$ describe two lines in the 2-D space $(\theta_{11},\theta_{22})$. Since the true value of θ_{11},θ_{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 θ and the risk $R(f)$. □

Clearly, the conditions for consistency in the asymmetric case are more restricted than in the symmetric case. However, situations such as in Proposition 13 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 1. A systematic way to do so may be obtained by noting that the identifiability equations

$$0 = \mathbb{P}_\theta(\hat{Y}_1, \ldots, \hat{Y}_k) - \mathbb{P}_\eta(\hat{Y}_1, \ldots, \hat{Y}_k) \quad \forall \hat{Y}_1, \ldots, \hat{Y}_k$$

is a system of polynomial equations in (θ, η). 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 [29]. 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 [103] for a survey of computational algorithms and software associated with systems of polynomial equations.

3.4.2 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 14. *Let $f_1, \ldots, f_k$ 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* (45) *is identifiable.*

Corollary 3. *Let $f_1, \ldots, f_k$ 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* (45) *is identifiable.*

Proof. Proving identifiability in the non-collaborative case proceeds by invoking Proposition 14 (whose proof is given below) with $k = 1$ separately for each regression model. The conditional independence assumption in Proposition 14 becomes redundant in this case of a single predictor, resulting in identifiability of $\mathbb{P}_{\theta_j}(\hat{Y}_j)$ for each $j = 1, \ldots, k$. □

Corollary 4. *Under the assumptions of Proposition 14 or Corollary 3 the unsupervised maximum likelihood estimator is consistent i.e.,*

$$P\left(\lim_{n\to\infty} \hat{\theta}_n^{mle}(\hat{Y}^{(1)}, \ldots, Y^{(n)}) = (\theta_1^{true}, \ldots, \theta_k^{true})\right) = 1.$$

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

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

Proof. Proposition 14 or Corollary 3 establish identifiability, which in conjunction with Proposition 11 completes the proof. □

Proof. **(of Proposition 14).**

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 $\mathbb{P}_{\theta_1}(\hat{Y}_1)$ in Equation (37). Substituting in it $\hat{Y}_1 = 0$ we get

$$\begin{aligned}
\mathbb{P}_{\theta_1}(\hat{Y}_1 = 0) &= \frac{1}{\theta_1\sqrt{2\pi(\tau^2 + \sigma_Y^2)}} \exp\left(\frac{\mu_Y^2}{2\sigma_Y^2}\left(\frac{\tau^2}{\sigma_Y^2 + \tau^2} - 1\right)\right) \\
\mathbb{P}_{\eta_1}(\hat{Y}_1 = 0) &= \frac{1}{\eta_1\sqrt{2\pi(\tau^2 + \sigma_Y^2)}} \exp\left(\frac{\mu_Y^2}{2\sigma_Y^2}\left(\frac{\tau^2}{\sigma_Y^2 + \tau^2} - 1\right)\right). \qquad (60)
\end{aligned}$$

The above expression leads to $\theta_1 \neq \eta_1 \Rightarrow \mathbb{P}_{\theta_1}(\hat{Y}_1 = 0) \neq \mathbb{P}_{\eta_1}(\hat{Y}_1 = 0)$ 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.,

$$\mathbb{P}_{(\theta,\theta_{k+1})}(\hat{Y}_1, \ldots, \hat{Y}_k, \hat{Y}_{k+1}) = \mathbb{P}_{(\eta,\eta_{k+1})}(\hat{Y}_1, \ldots, \hat{Y}_k, \hat{Y}_{k+1})\ \forall \hat{Y}_j \in \mathbb{R}\ j = 1, \ldots, k+1 \quad (61)$$

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 (61) with respect to $\hat{Y}_{k+1}$ which leads to a contradiction to our assumption that identifiability holds for k

$$\begin{aligned}
\int_{-\infty}^{\infty} \mathbb{P}_{(\theta,\theta_{k+1})}(\hat{Y}_1, \ldots, \hat{Y}_k, \hat{Y}_{k+1}) d\hat{Y}_{k+1} &= \int_{-\infty}^{\infty} \mathbb{P}_{(\eta,\eta_{k+1})}(\hat{Y}_1, \ldots, \hat{Y}_k, \hat{Y}_{k+1}) d\hat{Y}_{k+1} \\
\mathbb{P}_{\theta}(\hat{Y}_1, \ldots, \hat{Y}_k) &= \mathbb{P}_{\eta}(\hat{Y}_1, \ldots, \hat{Y}_k). \qquad (62)
\end{aligned}$$

In case (b) $\theta = \eta$ and $\theta_{k+1} \neq \eta_{k+1}$. Substituting $\hat{Y}_1 = \cdots = \hat{Y}_{k+1} = 0$ in (61) (see (46) for a derivation) we have

$$\mathbb{P}_{(\theta,\theta_{k+1})}(\hat{Y}_1 = 0, \ldots, \hat{Y}_{k+1} = 0) = \mathbb{P}_{(\eta,\eta_{k+1})}(\hat{Y}_1 = 0, \ldots, \hat{Y}_{k+1} = 0) \tag{63}$$

or

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

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

3.5 Asymptotic Variance of $\hat{\theta}_n^{mle}$ and $\hat{R}$

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

$$J(\theta) = \mathsf{E}_{\mathbb{P}_\theta}\{\nabla \log \mathbb{P}_\theta(\hat{Y})(\nabla \log \mathbb{P}_\theta(\hat{Y}))^\top\} \tag{64}$$

with $\nabla \log \mathbb{P}_\theta(\hat{Y})$ represents the $r \times 1$ gradient vector of $\log \mathbb{P}_\theta(\hat{Y})$ with respect to θ. Stated more formally, we have the following convergence in distribution as $n \to \infty$ [43]

$$\sqrt{n}\,(\hat{\theta}_n^{\text{mle}} - \theta_0) \rightsquigarrow N(0, J^{-1}(\theta^{\text{true}})). \tag{65}$$

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

In the case of classification considering (30) with $k = 1$ and $\mathcal{Y} = \{1, 2\}$ it can be shown that

$$J(\theta) = \frac{\alpha(2\alpha-1)^2}{(\theta(2\alpha-1) - \alpha + 1)^2} - \frac{(2\alpha-1)^2(\alpha-1)}{(\alpha - \theta(2\alpha-1))^2} \tag{66}$$

where $\alpha = \mathbb{P}(Y = 1)$. As Figure 14 (right) demonstrates, the asymptotic accuracy of the MLE (as indicated by J) tends to increase with the degree of non-uniformity of $\mathbb{P}(Y)$. Recall that since identifiability fails for a uniform $\mathbb{P}(Y)$ the risk estimate under a uniform $\mathbb{P}(Y)$ is not consistent. The above derivation (66) is a quantification of that fact reflecting the added difficulty in estimating the risk as we move closer to a uniform label distribution $\alpha \to 1/2$. The dependency of the asymptotic accuracy on θ^{true} is more complex, tending to favor θ^{true} values close to 1 or 0.5. Figure 14 (left) displays the empirical accuracy of the estimator as a function of $\mathbb{P}(Y)$ and θ^{true} and shows remarkable similarity to the contours of the Fisher information (see Section 4.6 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 $\theta^{\text{true}}, \mathbb{P}(Y)$. 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 [43] to $\hat{R}(f_j) = g_j(\hat{\theta}^{\text{mle}})$, $j = 1, \ldots, k$ and (65) we have

$$\sqrt{n}(\hat{R}(f) - R(f)) \rightsquigarrow N\left(0, \nabla g(\theta^{\text{true}}) J(\theta^{\text{true}}) \nabla g(\theta^{\text{true}})^{\top}\right) \tag{67}$$

where $R(f), \hat{R}(f)$ are the vectors of true risk and risk estimates for the different predictors $f_1, \ldots, f_k$ and $\nabla g(\theta^{\text{true}})$ is the Jacobian matrix of the mapping $g = (g_1, \ldots, g_k)$ evaluated at θ^{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 (65)-(66)

$$\sqrt{n}(\hat{R}(f) - R(f)) \rightsquigarrow N\left(0, \left(\frac{\alpha(2\alpha-1)^2}{(\theta(2\alpha-1)-\alpha+1)^2} - \frac{(2\alpha-1)^2(\alpha-1)}{(\alpha-\theta(2\alpha-1))^2}\right)^{-1}\right). \quad (68)$$

3.6 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 $\frac{\partial \ell}{\partial \theta_j}$is

$$\sum_{i=1}^{n} \frac{\sum_{Y^{(i)}} \mathbb{P}(Y^{(i)}) \sum_{r:\beta_{ri}=0} \sum_{\hat{Y}_r^{(i)}} \prod_{p\neq j} h_{pi} (I(\hat{Y}_j^{(i)} = Y^{(i)}) - \theta_j)((l-1)\theta_j)^{I(\hat{Y}_j^{(i)}=Y^{(i)})-1}(1-\theta_j)^{-I(\hat{Y}_j^{(i)}=Y^{(i)})}}{\sum_{Y^{(i)}} \mathbb{P}(Y^{(i)}) \sum_{r:\beta_{ri}=0} \sum_{\hat{Y}_r^{(i)}} \prod_{p=1}^{k} h_{pi}} \quad (69)$$

where

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

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

$$\theta^{(t+1)} = \arg\max_{\theta} \sum_{i=1}^{n} \sum_{Y^{(i)}} \sum_{r:\beta_{ri}=0} \sum_{\hat{Y}_r^{(i)}} q^{(t)}(\hat{Y}_r^{(i)}, Y^{(i)}) \sum_{j=1}^{k} \log \mathbb{P}_j(\hat{Y}_j^{(i)}|Y^{(i)}) \quad (70)$$

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

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

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 (49) reverts to (45), 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.

3.7 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 ℓ_1 error as a metric

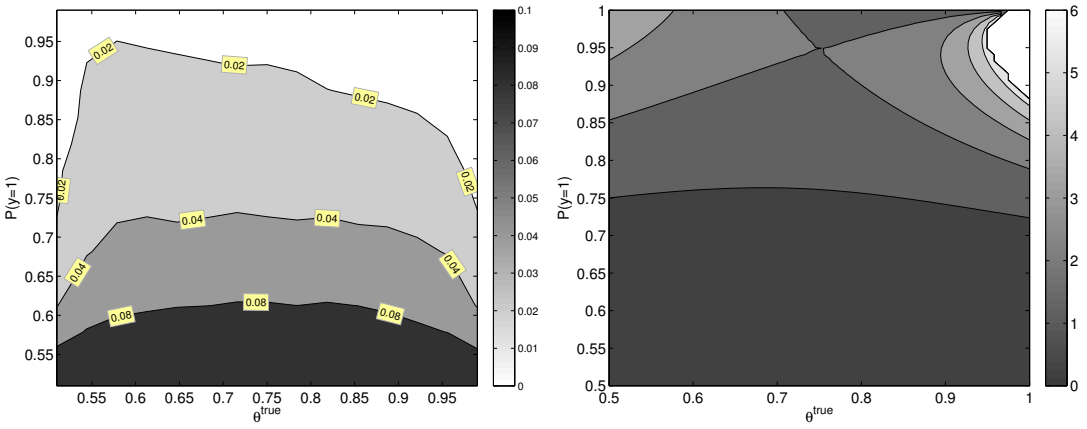


Figure 14: Left: Average value of $|\hat{\theta}_n^{\text{mle}} - \theta^{\text{true}}|$ as a function of θ^{true} and $\mathbb{P}(Y = 1)$ for $k = 1$ classifier and $n = 500$ (computed over a uniform spaced grid of 15×15 points). The plot illustrates the increased accuracy obtained by a less uniform $\mathbb{P}(Y)$. Right: Fisher information $J(\theta)$ for $k = 1$ as a function of θ^{true} and $\mathbb{P}(Y)$. The asymptotic variance of the estimator is $J^{-1}(\theta)$ which closely matches the experimental result in the left panel.

that measures the estimation quality

$$\text{mae}(\hat{\theta}^{\text{mle}}, \theta^{\text{true}}) = \frac{1}{k} \sum_{i=1}^{k} \left| \theta_i^{\text{true}} - \hat{\theta}_i^{\text{mle}} \right|. \tag{71}$$

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 14 (left) we display $\text{mae}(\hat{\theta}^{\text{mle}}, \theta^{\text{true}})$ for classification with $k = 1$ as a function of θ^{true} and $\mathbb{P}(Y)$ for $n = 500$ simulated data points. The estimation error, while overall relatively small, decays as $\mathbb{P}(Y)$ diverges from the uniform distribution. The dependency on θ^{true} indicates that the error is worst for θ^{true} around 0.75 and it decays as $|\theta^{\text{true}} - 0.75|$ increases with a larger decay attributed to higher θ^{true}. These observations are remarkably consistent with the developed theory as Figure 14 (right) shows by demonstrating the value of the inverse asymptotic variance $J(\theta)$ which agrees nicely with the empirical measurement in the left panel.

Figure 15 (left) contains a scatter plot contrasting values of θ^{true} and $\hat{\theta}^{\text{mle}}$ for $k = 1$ classifier and $\mathbb{P}(Y = 1) = 0.8$. The estimator was constructed based on 500 simulated

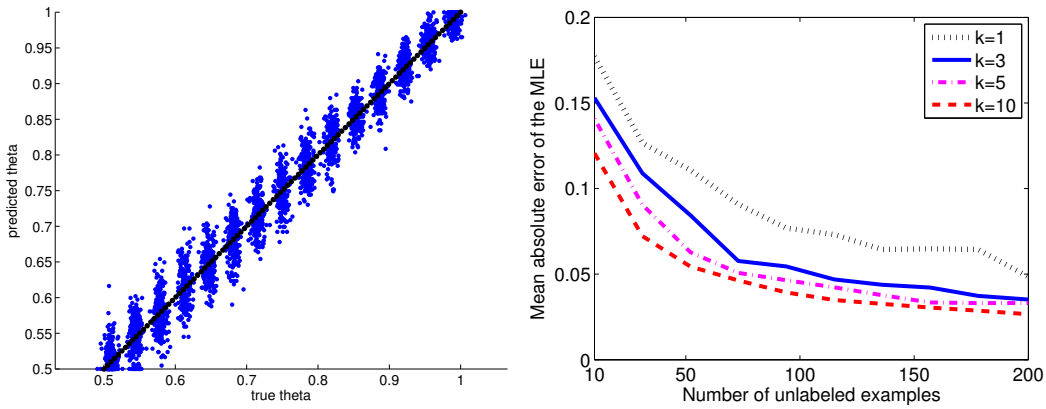


Figure 15: Left: Scatter plot contrasting the true and predicted values of θ in the case of a single classifier $k = 1$, $\mathbb{P}(Y = 1) = 0.8$, and $n = 500$ unlabeled examples. The displayed points were perturbed for improved visualization and the striped effect is due to empirical evaluation over a discrete grid of θ^{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 ($\theta_i^{\text{true}} = \mathbb{P}(Y = 1) = 0.75$) in the collaborative case. The estimation error decreases as more classifiers are used due to the collaborative nature of the estimation process.

data points. We observe a symmetric Gaussian-like distribution of estimated values $\hat{\theta}^{\text{mle}}$, conditioned on specific values of θ^{true}. This is in agreement with the theory predicting an asymptotic Gaussian distribution for the mle, centered around the true value θ^{true}. A similar observation is made in Figure 16 (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 θ^{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 θ^{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 15 and 16 (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 $\mathbb{P}(Y = 1) = 0.75 = \theta_i^{\text{true}}, \forall i$. In regression, the data was sampled from the regression equation with $\theta_i^{\text{true}} = 1$ and $\mathbb{P}(Y) = N(0, 1)$. In both cases, the mae error decays with n as expected from the consistency proof and with k as a result of the

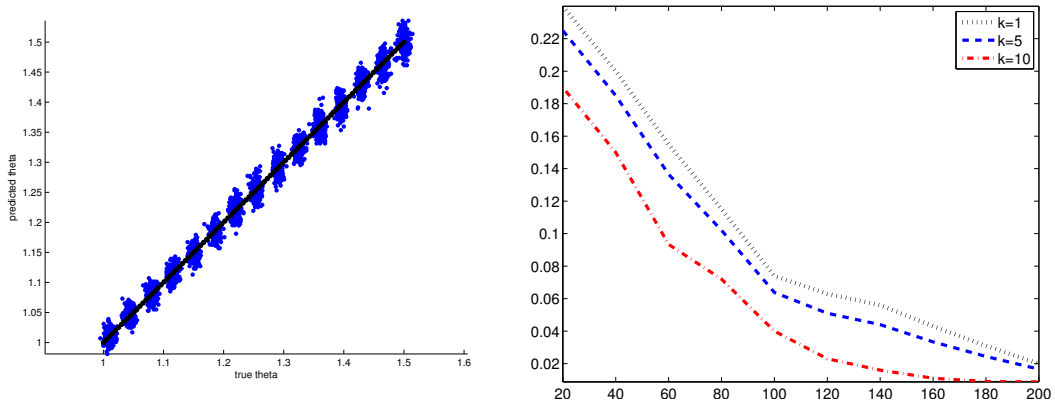


Figure 16: Left: Scatter plot contrasting the true and predicted values of θ in the case of a single regression model $k = 1$, $\sigma_Y = 1$, and $n = 1000$ unlabeled examples. The displayed points were perturbed for improved visualization and the striped effect is due to empirical evaluation over a discrete grid of θ^{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.

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 17. 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 17) demonstrating the improvement of the collaborative process.

We compare in Figure 18 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 as well as the supervised version especially in general. Nevertheless,

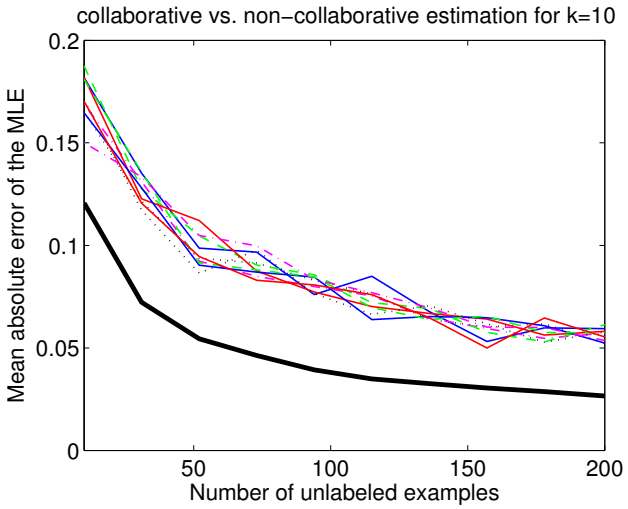


Figure 17: Comparison of collaborative and non-collaborative estimation for $k = 10$ classifiers. $\text{mae}(\hat{\theta}^{\text{mle}}, \theta^{\text{true}})$ as a function of n is reported for $\theta_i^{\text{true}} = 0.75\ \forall k_i$ and $\mathbb{P}(Y = 1) = 0.75$. The colored lines represent the 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.

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 19 we report the effect of misspecification of the marginal $\mathbb{P}(Y)$ 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 20 shows the mean prediction accuracy for the unsupervised predictor combination scheme in (25) 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. The

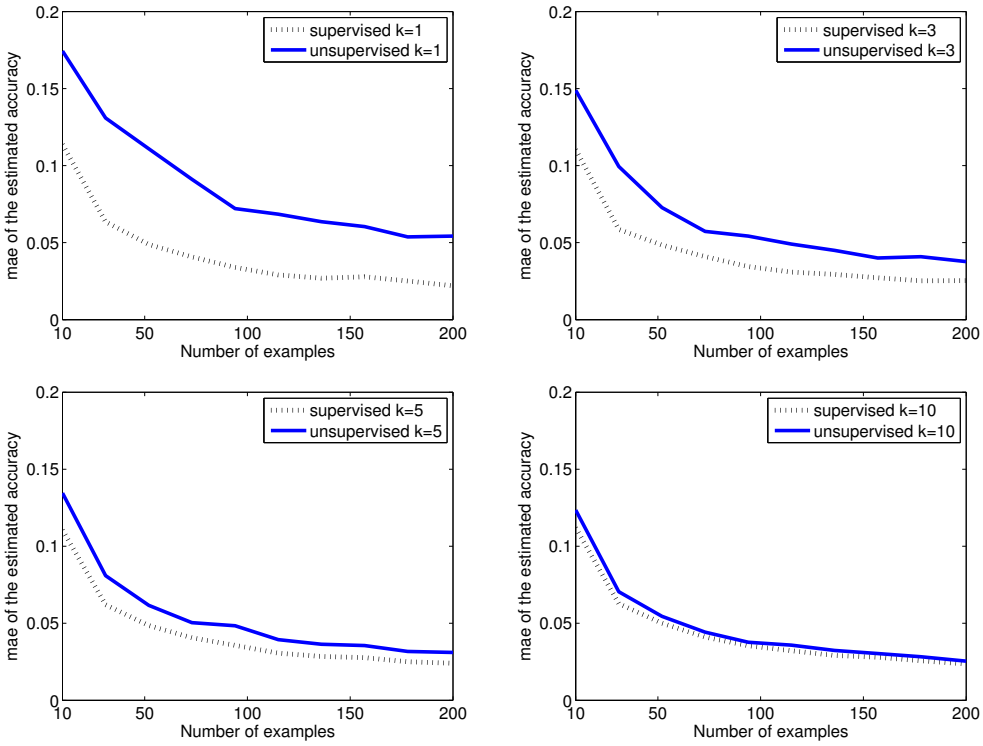


Figure 18: Comparison of supervised and unsupervised estimation for different values of classifiers with $k = 1, 3, 5, 10$. Supervised estimation uses the true labels to determine the accuracy of the classifiers whereas in the unsupervised case the estimation proceeds according to the collaborative estimation framework. Despite the fact that the supervised case uses labels the unsupervised framework reaches similar levels by increasing the number of classifiers.

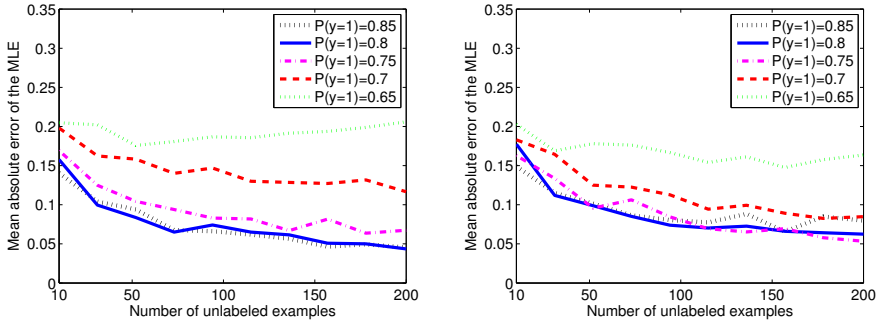


Figure 19: The figure compares the estimator accuracy assuming that the marginal $\mathbb{P}(Y)$ is misspecified. The plots draw $\text{mae}(\hat{\theta}^{\text{mle}}, \theta^{\text{true}})$ as a function of n for $k = 1$ and $\theta^{\text{true}} = 0.75$ when $P^{\text{true}}(Y = 1) = 0.8$ (left) and $P^{\text{true}}(Y = 1) = 0.75$ (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)

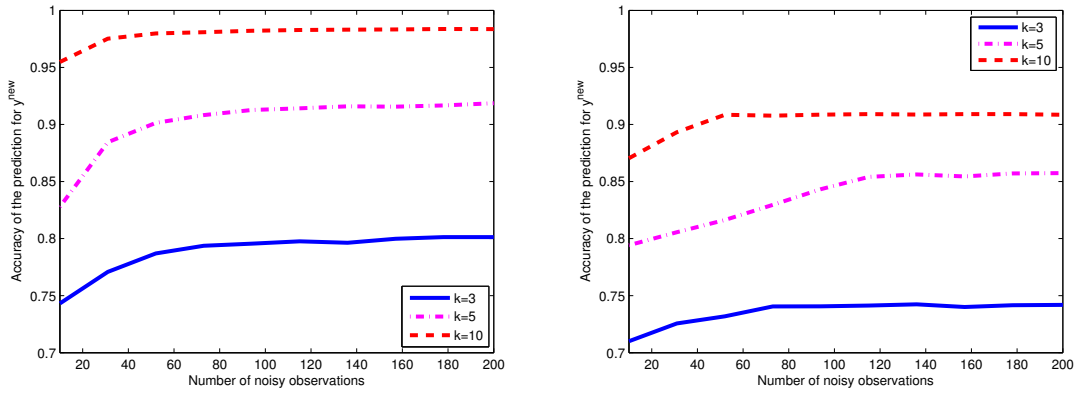


Figure 20: Mean prediction accuracy for the unsupervised predictor combination scheme in (25) for synthetic data. The left panel displays classification accuracy and the right panel displays the regression accuracy as measured by $1 - \frac{1}{m}\sum_{i=1}^{m}(\hat{Y}_i^{\text{new}} - y_i^{\text{new}})^2$. The graphs show that in both cases the accuracy increases with k and n in accordance with the theory and the risk estimation experiments.

parameter θ_i^{true} was chosen uniformly in the range $(0.5, 1)$, and $\mathbb{P}(Y = 1) = 0.75$ for classification and $\theta_i^{\text{true}} = 0.3$, $\mathbb{P}(Y) = N(0, 1)$ in the case of regression.

We also experimented with the natural language understanding dataset introduced in [97]. 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 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

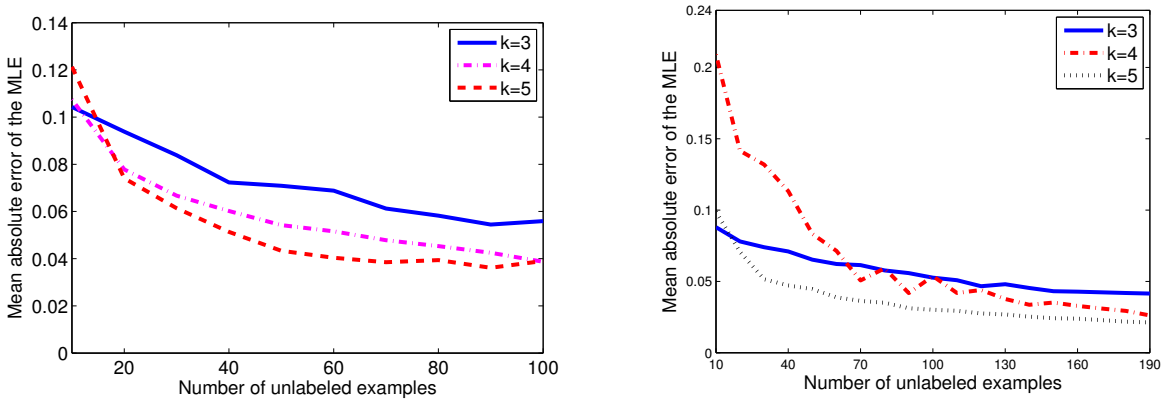


Figure 21: $\text{mae}(\hat{\theta}^{\text{mle}}, \theta^{\text{true}})$ as a function of n for different number of annotators k on RTE (left) and TEMP (right) datasets. Left: $n = 100$, $\mathbb{P}(Y = 1) = 0.5$ and $\theta^{\text{true}} = \{0.85, 0.92, 0.58, 0.5, 0.51\}$. Right: $n = 190$, $\mathbb{P}(Y = 1) = 0.56$ and $\theta^{\text{true}} = \{0.93, 0.92, 0.54, 0.44, 0.92\}$. The classifiers were added in the order specified.

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.

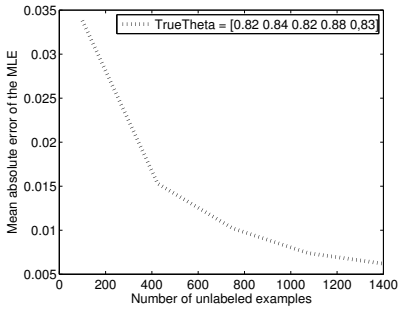
In Figure 21 we display $\text{mae}(\theta^{\text{true}}, \hat{\theta}^{\text{mle}})$ for these datasets as function of n for different values of k. These plots generated from real-world data show similar trend to the synthetic experiments. The estimation errors decay to 0 as n increases and generally 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

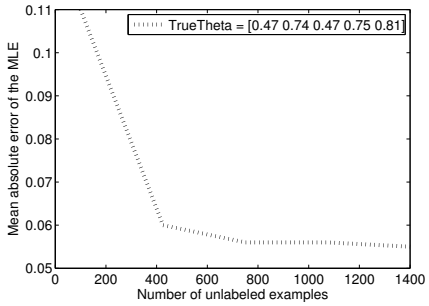
generated by [19]. Ringnorm is a 2-class artificial dataset with 20 dimensions where each class is drawn from a multivariate normal distribution. One class has zero mean and a covariance $\Sigma = 4I$ where I is the identity matrix. The other class has unit covariance and a mean $\mu = (\frac{2}{\sqrt{20}}, \frac{2}{\sqrt{20}}, \ldots, \frac{2}{\sqrt{20}})$. 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) [116, 62] 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 22 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 22(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 22(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 3.5 (recall the Fisher information contour plot).

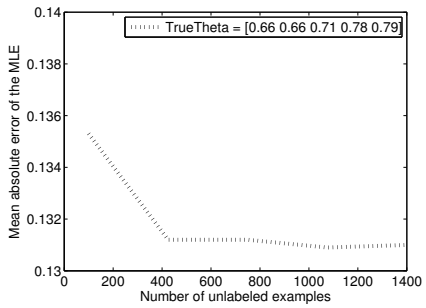
Our experiments thus far assumed the symmetric noise model (30). Despite it not being always applicable for real world data and classifiers, it did result in good estimation accuracy in some of the cases described thus far. However, in some cases this assumption is grossly violated and the more general noise model is needed (28). For this reason, we conducted two experiments using real world data assuming the



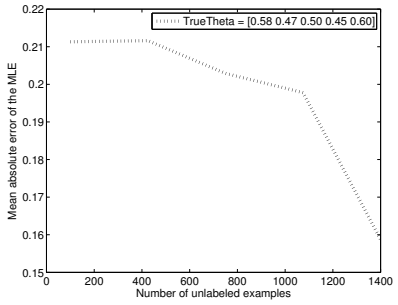
(a) Strong classifiers



(b) A mixture of strong and weak classifiers



(c) Mostly weak classifiers



(d) Very weak classifiers

Figure 22: $\mathrm{mae}(\theta^{\mathrm{true}}, \hat{\theta}^{\mathrm{mle}})$ as a function of the test set size on the Ringnorm dataset. $\mathbb{P}(Y = 1) = 0.47$, and θ^{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	**0.04**	**0.04**	**0.08**	**0.06**	**0.006**
collaborative	0.10	0.10	0.09	0.08	n/a

Figure 23: $\text{mae}(\hat{\theta}^{\text{mle}}, \theta^{\text{true}})$ for the domain adaptation ($n = 1000$, $\mathbb{P}(Y = 1) = 0.75$) and 20 newsgroup ($n = 15,000$, $\mathbb{P}(Y = 1) = 0.05$ 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.

more general (28).

The first experiment concerned domain adaptation [18] 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 $\mathbb{P}(Y = 1) = 0.75$. The task was to estimate the error rates of classifiers (linear SVM [116, 62]) 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 23 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.

In the second experiment using (28) we estimated the risk (non-collaboratively) of 20 one vs. all classifiers (trained to predict one class) on the 20 newsgroup data [65]. 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 23).

3.8 Discussion

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 [34], bootstrap [35], and others [50], our approach is fully unsupervised and thus able to use vast collections of unlabeled data. Other related work includes [96] and [92] 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, [96] and [92] fail to address estimating the risks of the predictors which is the main focus of our work.

We prove statistical consistency in the unsupervised case and derive the asymptotic variance. Our experiments on synthetic data demonstrate the effectiveness of the framework and verify the theoretical results. Experiments on real world data show robustness to underlying assumptions. The framework may be applied to estimate additional quantities in an unsupervised manner, including noise level in noisy communication channels [28] and error rates in structured prediction problems.

CHAPTER IV

LANDMARK SELECTION METHOD FOR MULTIPLE OUTPUT PREDICTION

4.1 Introduction

Conditional modeling $\mathcal{X} \mapsto \mathcal{Y}$ is a central problem in machine learning. Specific cases include classification, where $\mathcal{Y}$ is a discrete random variable, and regression, where $\mathcal{Y}$ is a continuous random variable. Much of the attention in recent years has focused on the case where $\mathcal{X}$ is a high dimensional vector ($\mathcal{X} \subset \mathbb{R}^d$). In this case, traditional statistical methods are inefficient due to overfitting. Proposed alternatives for high dimensional $X \in \mathbb{R}^d$ include feature selection and regularized models.

We consider, instead, the case of a high dimensional Y, where X is either low dimensional or high dimensional. The baseline approach in this case is to independently construct models $X \mapsto Y_i \in \mathbb{R}$ for $i = 1, \ldots, k$ (assuming Y is a k-dimensional real vector). This approach has the advantage of drawing from a wide variety of available single output models, including linear and non-linear regression, logistic regression, and support vector machines. The main disadvantage is that the independent models do not take advantage of a likely correlation between the dimensions of Y. Incorporating this correlation becomes especially important when the dimensionality of Y is higher or of similar order to the dimensionality of X.

Our approach is based on selecting a small subset $L \subset \{1, \ldots, k\}$ of the dimensions of Y, and constructing two models:

$$X \mapsto Y_L \tag{72}$$

$$Y_L \mapsto Y, \tag{73}$$

where we use the standard notation $Y_L = \{Y_i : i \in L\}$. We thus have three problems: selecting the subset L, estimating (72), and estimating (73).

Specifically, we estimate model (73) in conjunction with selecting L via least-squares regression with group Lasso based hierarchical regularization. The precise model (72) varies, based on whether Y is discrete or continuous. It may be any low-dimensional multiple output model, such as multilabel logistic regression and SVM, or multiple linear regression. If the dimensionality of X is high, regularization for model (72) is also necessary.

The underlying assumption of our model is that there exists a subset L of the dimensions of Y, called landmark variables, such that the remaining dimensions of Y may be expressed as a noisy linear combination $Y = AY_L + \epsilon$, with sparse coefficients. Several practical data sets exhibit such a kind of relationship. One example is the prediction of future stock prices Y from current stock prices X. The relationship $Y = AY_L + \epsilon$ is motivated by the identical trends of stock prices of multiple companies with a similar business model, or of multiple investment banks with similar holding portfolio. This phenomenon has been well documented in finance under the term cointegration. Another example is the classification of images (X) depending on what objects appear or do not appear in them (Y). Obviously, some objects tend to appear or to not appear simultaneously, such as sky and tree, or car and road.

The cardinality s of the subset L is typically orders of magnitude lesser than the actual dimension of the output space making the method scale well to ultra-high dimensional outputs. For example, the naive one vs. all method requires $O(k)$ independent models that need to be learnt from the data, whereas the number of subproblems selected in the proposed approach scales at the rate of $O(s)$. Assuming $s \ll k$ we see that there is a huge advantage in terms of number of subproblems selected.

We report in this chapter experimental results for classification and regression on

multiple datasets. Based on our experimental study, we conclude that our model outperforms the one vs. all approach as well as several sophisticated multiple output prediction methods.

4.2 Related Work

Several methods have been proposed for multi-output prediction both in regression and classification setting. In the regression setting, most approaches have focused on penalization of the regression matrix or input space sharing. For example [58] introduced low-rank penalization of the regression matrix, which was analyzed in [85] in the low-dimensional setting. Recent work focused on analyzing penalized regression in high dimensions [88]. An alternative approach that is directly applicable to multi-output prediction is group lasso [126]. Though these methods are popular and widely applicable, they do not directly model correlations in the output dimensions, which can be used to reduce the complexity of the problem. A notable exception is the curds and whey method [20] which uses shrinkage techniques in output space to reduce prediction error.

In the classification setting, the popular approach of one-vs-all was proposed by several researchers (see [86] for a discussion). This method ignores the dependencies between the different dimensions of Y, and is inefficient when Y is high dimensional. A summary of improvements over the one-versus-all method is available in [113]. Alternative approaches assume a class hierarchy on the output space [26], graph structure on the output space [117] and joint feature extraction from output and input spaces in large margin setting [112].

A chapter related to our proposed method is [54], which consider multi-label prediction in a sparse high-dimensional output space. Their proposed method for multi-label classification is to randomly project Y and construct a regression model on the reduced subspace. There are two significant differences between this chapter

and our approach: (i) our approach uses data-dependent transformation, rather than a random projection, and (ii) our approach selects a subset of the dimensions of Y that contributes to computational efficiency, statistical analysis, and is in line with some practical scenarios (see previous section). Furthermore, the approach by [54] might not be applicable in the regression setting, as output sparsity assumption does not hold for regression in practice.

Recently proposed variations on [54] include [105] that propose to reduce the dimensionality of the output space by PCA, and [14] that propose to reduce the label space by preserving a graph structure hierarchy on Y. While these methods are sub-linear, they still project on to a low-dimensional real subspace, and hence they do not guarantee that the problem in the reduced subspace is easier than the original problem.

Our approach also has a close connection to sparse PCA [132]. Two significant differences are: (i) sparse PCA is generally applied to the covariates X rather than Y, and (ii) our focus is on identifying the landmarks L and the relationship $Y_L \mapsto Y$ rather than estimating the principal components themselves.

4.3 The landmark selection method:

With a slight abuse of notation, we denote the data matrices, containing n labeled samples, by $X \in \mathbb{R}^{n \times d}$ and $Y \in \mathbb{R}^{n \times k}$.

4.3.1 Step 1: Selecting the landmark set L and modeling (73)

A convenient way to select the set of landmark dimensions L, and to model (73) simultaneously is the following regularized least squares regression model

$$\hat{A} = \underset{A \in \mathbb{R}^{k \times k}}{\arg\min} \|Y - YA\|_F^2 + \lambda_1 \|A\|_{1,2} + \lambda_2 \|A\|_1 \tag{74}$$

where

$$\|A\|_F \stackrel{\text{def}}{=} \sqrt{\text{tr}(A^\top A)}$$
$$\|A\|_{1,2} \stackrel{\text{def}}{=} \sum_{i=1}^{k} \sqrt{\sum_{j=1}^{k} A_{i,j}^2}$$
$$\|A\|_1 \stackrel{\text{def}}{=} \sum_{i=1}^{k} \sum_{j=1}^{k} |A_{ij}|.$$

The first term in (74) is the least squares empirical risk that is standard in linear regression models. Obviously, the identity $A = I$ minimizing that term constitutes a trivial solution that is ineffective when Y is high dimensional. The second and third terms in (74) promote a "small" A and thus prevent the estimated model to be the trivial minimizer I of the first term.

Much like group lasso, the second term in (74) enforces joint group sparsity across the rows of A. To see this note that $\|A\|_{1,2}$ is the L_1 norm of the L_2 norms of the individual rows. Due to the sparsity promoting nature of the L_1 minimizer, $\hat{A}$ will have only a few rows that are not identically zero. The resulting effect is the selection of landmark dimensions Y_L where L corresponds to the non-zero rows. We thus have that the first two terms in (74) simultaneously select the landmark dimensions L, and model $Y_L \mapsto Y$. The third term $\|A\|_1$ promotes sparsity within the coefficients of the model $Y_L \mapsto Y$. This additional sparsity assumption reduces the prediction risk when Y is high dimensional.

The regularization parameter λ_1 controls the number of landmark output dimensions. The regularization parameter λ_2 controls the sparsity of the model $Y_L \mapsto Y$. Both λ_1 and λ_2 should increase with k. When the landmark assumption holds and there exists a landmark set L^* such that Y is a noisy sparse linear combination of y_{L^*}, the row sparsity pattern of $\hat{A}$ should coincide with L^* (assuming an appropriate selection of λ_1, λ_2). As λ_1/λ_2 increases, the group sparsity constraints become dominant implying that each dimension of Y depends on all of the dimensions of Y_L. As λ_1/λ_2

decreases, $\hat{A}$ tends to be more sparse within groups, implying that the dimensions of Y are sparse linear combinations of the Y_L.

From a practical point of view, with a proper selection of the regularization parameters λ_1, λ_2 (for example using cross-validation), the model (74) is quite flexible. It allows handling situations involving large landmark sets L and small landmark sets L, and high or low sparsity for the model $Y_L \mapsto Y$. Empirically, the dependence on the precise value of λ_1, λ_2 is robust, as small variations in λ_1, λ_2 do not substantially change the predicted values.

Handling non-linear output relationship: In order to select and learn non-linear relationships between the outputs and the landmarks, one could use functional joint sparsity models with $Ł_1/L_\infty$ constraints as proposed by [69] or with $L_1 + Ł_1/L_2$ constraints (appropriately defined on a function space). With this change in step 1, the proposed approach could be used to handle non-linear relationships between the outputs, making the proposed method more flexible. Developing concrete algorithms and analysis for this setting is left as future work.

4.3.2 Step 2: Estimating (72)

Once the landmark outputs L are identified, we can proceed with fitting model (72). In the case of continuous Y (regression), model (72) can be estimated using a using multivariate regression model. In the case of a discrete Y (classification), a one vs. all classifier may be used for $X \mapsto Y_i$, $i = 1, \ldots, s$, or alternatively a multiple output classifier may be used for $X \mapsto Y_L$. Examples include support vector machines and log-linear models. From a statistical perspective, when Y is high dimensional the reduction in the number of estimated parameters from kd to sd (in the regression setting) where $s \ll k$, contributes to lower prediction risk. If the dimensionality of X is also high, the models $X \mapsto Y_L$ or $X \mapsto Y_i$ should use careful feature selection or regularization to avoid overfitting.

Algorithm 3 Landmark selection method

Input: data $\{(X^{(1)}, Y^{(1)}), \ldots, (X^{(n)}, Y^{(n)})\}$ in the form of $X \in \mathbb{R}^{n \times d}$ and $Y \in \mathbb{R}^{n \times k}$
Step 1: Simultaneously find the landmark set L and solve the optimization problem in step 1 to obtain the model $Y_L \to Y$ and estimate $\hat{A}$.
Step 2: Estimate the model $X \to Y_L$ using independent models for each component of Y_L or using multiple-output classification or regression algorithms.
Step 3: Given a new test point X, estimate Y by (75)-(76).

4.3.3 Step 3: Prediction

In many cases, a statistical model for (72) provides not only point estimates, but also a full probabilistic model $\mathbb{P}(Y_L|X)$. Similarly, a statistical model for (73) provides a full probabilistic model for $\mathbb{P}(Y|Y_L)$. The implied model

$$\mathbb{P}(Y|X) = \mathbb{P}(Y|Y_L)\mathbb{P}(Y_L|X)$$

suggests the following procedure for predicting Y from X

$$Y_L^* = \arg\max_{Y_L} p(Y_L|x) \tag{75}$$

$$y_{L^c}^* = \arg\max_{Y_{L^c}} \int \mathbb{P}(Y|Y_L)\mathbb{P}(Y_L|X)\, dY_L. \tag{76}$$

An alternative to (76) is to use the following approximation

$$\arg\max_Y \mathbb{P}(Y|X) \approx \arg\max_Y \mathbb{P}\left(Y|\arg\max_{Y_L} \mathbb{P}(Y_L|X)\right).$$

In other words, given a new test sample X, we predict Y_L using the model from step 2, and then estimate Y_{L^c} using the model from step 1, operating on the predicted Y_L. In the case of classification, we follow standard practice and set the components of Y to 1 if the corresponding prediction of model (73) is greater than 0.5 and to 0 if it lesser than 0.5. Finally the outputs are concatenated and they represent the prediction for the given sample X. Algorithm 1 summarizes this procedure.

4.4 Theory

In this section, we give a brief theoretical analysis of the proposed approach in the regression setting highlighting the advantage of the proposed approach. We assume

that there exist a true landmark subset L^* and provide conditions under which it could be recovered consistently. Specifically, following the analysis developed in [78] for random design linear regression with group Lasso regularization, we can get a lower bound on the number of samples needed for recovering the support of the subset L^* of the landmark labels. For simplicity, we consider the regression setting with the assumption that $\lambda_2 = 0$.

We assume that Y consists of i.i.d. rows sampled from $N(0, \Sigma)$. This distribution could in fact be any sub-Gaussian distribution (which includes any bounded random variable for example the Bernoulli random variable) for which a similar analysis could be carried out. We make the following assumption on the the covariance matrix Σ: (1) there exists $\rho_{min} > 0$ and $\rho_{max} < \infty$ such that all the eigenvalues of the $s \times s$ covariance matrix Σ_s of the the landmark output $Y_L \in \mathbb{R}^s$ are contained in the closed interval $[\rho_{min}, \rho_{max}]$, (2) *mutual incoherence:* there exist a incoherence parameter $\gamma \in (0, 1]$ such that $\|\Sigma_{S^cS^c}(\Sigma_{SS})^{-1}\|_\infty \leq 1 - \gamma$ and (3) *self-incoherence:* there exists $D_{max} < \infty$ such that $\|(\Sigma_{SS})^{-1}\|_\infty \leq D_{max}$. Note that these are standard conditions assumed for support recovery results in the modern sparse recovery analysis. Condition (1) is needed to prevent over-dependency between the landmark outputs. Conditions (2) and (3) are necessary conditions for model selection consistency of sparse recovery problems. For example, several classes of matrices, for example Toeplitz matrices, tree-structured matrices and bounded off-diagonal matrices are shown to satisfy the above conditions [129]. In the absence of these conditions, landmark recovery might fail even with arbitrarily large training set.

We also make the following assumption on the regression matrix. Let $a_{min} \stackrel{\text{def}}{=} \min_{i \in L} \|A_i\|_2$ where A_i denote the i^{th} non-zero row of the matrix A. We denote $A^s \in \mathbb{R}^{s \times k}$ to be the subset of the matrix A with non-zero rows, $\zeta(A_s) \in \mathbb{R}^{s \times k}$ to be the row normalized matrix, and

$$\phi(A) \stackrel{\text{def}}{=} \lambda_{max}\big(\zeta(A_s)^\top (\Sigma_{SS})^{-1} \zeta(A_s)\big).$$

This quantity characterizes the amount of overlap that could be captured given the output samples. Note that the support overlap function $\phi(A)$ satisfies

$$\frac{s}{\rho_{max}K} \leq \phi(A) \leq \frac{s}{\rho_{min}}$$

for any Y that satisfies assumption (1).

Proposition 15. *Consider the label matrix Y with rows i.i.d. drawn from $N(0, \Sigma)$ satisfying assumptions (1)-(3), suppose that a_{min}^2 decays no more slowly than $f(k)\min\{\frac{1}{s}, \frac{1}{\log(k-s)}\}$ for some function $f(k)$ such that $f(k)/s \to 0$ and $f(k) \to \infty$. Then, as long as $n > C'\rho_{max}\phi(A^*)\log(k-s)$, we have with probability greater than $1 - c_1\exp(c_2 \log s)$: (1) the optimization problem in 74 (with $\lambda_2 = 0$) has a unique solution when $\lambda_1 = \sqrt{\frac{f(k)\log k}{n}}$ and (2) the row support specified by the unique solution of the optimization problem 74 is equal to the row support of the true model.*

Proof. The proof follows from the corresponding proof in [78]. □

The main consequence of the above proposition is that if there exist a set of landmark variable L^* in the output space, the sample complexity is of *logarithmic order* in the original dimension of the output space k. Using sub-Gaussian assumptions on the label matrix, analogous conditions for classification are possible.

Following [85] we note that for a matrix regression problem $y = \Theta x + \epsilon$ with $\Theta \in \mathbb{R}^{m_1 \times m_2}$, the Frobenious norm error rate (with n samples, unit noise variance and no assumption on the regression matrix)

$$\|\hat{\Theta} - \Theta\|_F^2 = O\left(\frac{m_1 m_2}{n}\right).$$

Since in our case the estimated matrix (72) (assuming linear regression model) is of the dimension $s \times d$, the error is of the order of $O(\frac{sd}{n})$ samples [85], much smaller than the classical setting without the landmark selection method of $O(\frac{kd}{n})$. In particular, when $s \ll k$, there is a significant gain in efficiency.

We conclude that the landmark method makes a structural assumption on the output space in order to facilitate regression in high dimensional setting ($n \ll kd$). Other methods, making a different set of structural assumptions (e.g., low-rank regression) try to achieve the same goal, but work under a different set of assumptions. Empirically, the landmark method works better than low-rank regression and group Lasso based multivariate regression on a variety of datasets (see Section 4.6).

4.5 Optimization procedure

Here, we provide the optimization procedure required to solve the optimization problem described in step 1. The spaRSA method, proposed recently in [120], is a solver for optimization problems of the form

$$\min_{a \in \mathbb{R}^p} f(a) + \lambda \phi(a)$$

where f is a convex loss function and ϕ is a convex regularizer. The main advantage of spaRSA is that when the regularizer is group separable, the problem decomposes over the group.

Using vectorization and block-diagonalization, it can be shown that (74) falls under this framework. Upon initial investigation, it appears that the block-diagonalization operation complicates the solver as it increases the size of the data matrix. However, we describe below a variation on spaRSA that works directly with the Y and A. A similar approach was used in [100] for the problem of collaborative dictionary learning with hierarchical penalty. The main advantage of the spaRSA procedure (that the problem decouples across groups) is still preserved and further in our case, each subproblem could be solved via thresholding.

In order to solve the optimization problem, the spaRSA procedure generates a sequence of updates that converges to the solution. We refer the reader to [120] for a complete description of the general procedure. In our case, we let $f(A_i)$ denote the reconstruction error (the squared loss in our case) for A_i (here and below we denote

the i-column of a matrix A as A_i) and define the matrix $U^{(t)} \in \mathbb{R}^{k\times k}$ whose i-column is given by

$$U_i^{(t)} = A_i^{(t)} - (1/\alpha^t)\nabla f(A_i^{(t)}).$$

The sequence of spaRSA updates that converge to the true solution is

$$A^{(t+1)} = \arg\min_{Z\in\mathbb{R}^{k\times k}} \|Z - U^{(t)}\|_F + \frac{\lambda_1}{\alpha^{(t)}}\|Z\|_F + \frac{\lambda_2}{\alpha^{(t)}}\|Z\|_1,$$

which is group separable into k independent problems as below:

$$A_i^{(t+1)} = \arg\min_{Z_i\in\mathbb{R}^{k}} \|Z_i - U_i^{(t)}\|_2 + \frac{\lambda_1}{\alpha^{(t)}}\|Z_i\|_2 + \frac{\lambda_2}{\alpha^{(t)}}\|Z\|_1.$$

The solutions for each of these sub-problems are available in closed form (similar to [100]) as follows:

$$A_i^{(t+1)} = \begin{cases} \max\{0, \|h\|_2 - \lambda_1\}h/\|h\|_2 & \text{if } \|h\|_2 > 0 \\ 0 & \text{if } \|h\|_2 = 0 \end{cases}$$

where $h_j = \text{ sign}(U_{i,j}^{(t)})\max\{0, |U_{i,j}^{(t)}| - \lambda_2\}$. The thresholding require operations that are linear in the dimensionality of the matrix Y. The above procedure is repeated until convergence to obtain the final solution that features row sparsity, and potential sparsity within rows as well.

4.6 Experiments

In this section, we compare our landmark selection method, which we refer to below as moplms, to alternative baselines on classification and regression problems. In our experiments we used code from [120] for performing the mixed norm penalty (group lasso and lasso) landmark selection. The regularization parameters were set by cross-validation.

4.6.1 Synthetic experiments

We conducted an experiment on synthetic regression data with $k = 500$ (dimensionality of Y), $d = 500$ (dimensionality of X). The number of landmark outputs s was

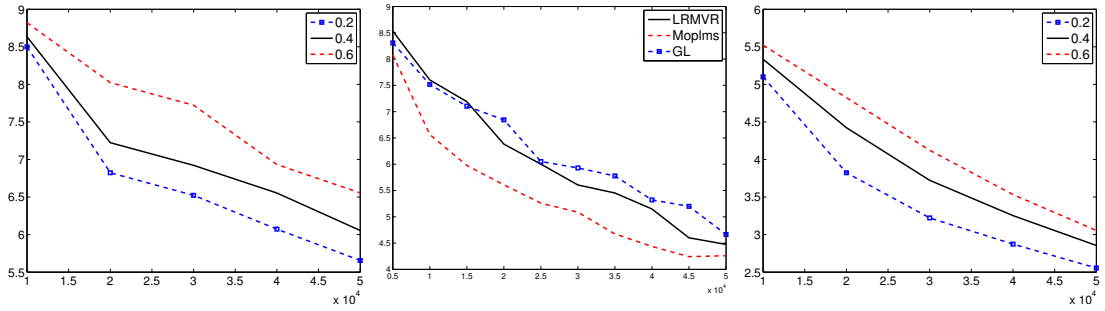


Figure 24: Left: MSE vs. sample size for synthetic regression data set. Middle: MSE vs. sample size for synthetic regression data set. Right: Hamming loss as a function of sample size for synthetic classification data set. The multiple curves represent different values of s/k.

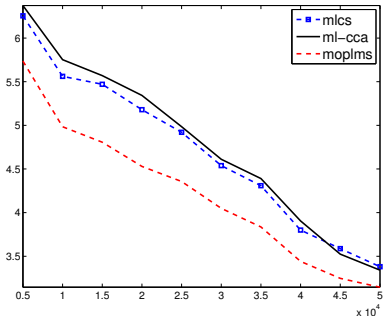


Figure 25: Hamming loss vs. sample size on synthetic classification data sets.

varied in the set $\{50, 100, 200\}$. The data was simulated from the above model, including the specified landmark outputs. Figure 24 (left) shows the plot of the test MSE prediction error as a function of the sample size for various values of the parameter s/k.

From section 5.4, we have that if the landmark output selection method is not used, with a linear regression model for $X \mapsto Y$,the Frobenious norm error between the true and estimated matrix scales as $O\big(\frac{kd}{n}\big)$. Where as with the landmark output assumption the error for model 72 scales as $O\big(\frac{sd}{n}\big)$. This benefit in the estimation error of the regression matrix is reflected in the MSE prediction error. Specifically, as s decreases, the sample complexity decreases. This phenomenon is especially important in high-dimensional cases, when there are fewer samples than the number of parameters to be estimated. We also compared the proposed approach to group-Lasso

and low-rank multivariate regression. Figure 24 (middle) shows the mse prediction error rate of the moplms method decays faster compared to the other methods.

We also experimented with synthetic classification data where Y is a 500 dimensional binary vector and the input $X \in \mathbb{R}^{500}$. Similar to the regression setting, the landmark outputs were first generated with $s \in \{50, 100, 200\}$ and the dependent outputs where generated as sparse linear combination of the landmark outputs. Figure 24 (right) shows the Hamming loss as a function of the sample size. The $X \mapsto Y_L$ model was collection of multiple one-vs-all SVMs. Similar to the regression case, the prediction error decays with the number of landmark outputs s/k. We further compare the proposed approach on synthetic data set against the following methods:

1. **One vs. all:** This is a standard base-line approach for multi-label classification, for e.g., [86].

2. **Multilabel compressive sensing (mlcs):** This approach was proposed in [54] where the label vector is projected to a random m dimensional sub-space followed by regression on the compressed subspace.

3. **Multi-label classification via canonical correlation analysis (ml-cca):** After performing canonical correlation analysis (CCA) on the input and output variables, a model is learned in the resulting subspace, followed by projection to the original label space.

From Figure 25, we note that the proposed approach has a better rate of decay of hamming loss compared to the other approaches. This phenomenon is further observed in the real world data sets as described in the next section.

We conducted an additional experiment to study the number of sub-problems selected. Specifically, we varied the number of sub-problems and the tuning parameters of mlcs, and noted the values achieving the lowest prediction error. We then trained moplms, gradually reducing the regularization parameter until the prediction error

matched that of mlcs. The two methods achieved identical prediction error with the following (mean) values of s/k: 0.45 (mlcs) and 0.30 (moplms), indicating moplms selected fewer sub-problems while achieving identical performance. Note, however, that mlcs always uses base regressors and moplms uses base classifiers.

4.6.2 Real-world data sets

4.6.2.1 Classification

We experimented with the following two multiple output classification datasets.

1. **del.icio.us** This dataset consists of data from del.icio.us, a social bookmarking site where webpages are labeled with multiple contextual tags. The data set contains about 16000 labeled web page and 983 unique labels. We follow the experimental setup followed in [54] and represent web page as a boolean bag-of-words vector, with the vocabulary chosen using a combination of frequency thresholding and χ^2 feature ranking, resulting in 500 features.
2. **Image data set.** This dataset contains 68000 images, with about 22000 unique word tags for each image. Following [54] we retained the 1000 most frequent labels. We represented each image via codes computed with a learned dictionary (of size 1024) via sparse coding [121]. Specifically, we densely sampled 10×10 patches from the image and computed sparse codes. Finally max-pooling was used to pool the codes obtained for the patches.

Note that we use thresholding to convert the real output to the binary form of the data. The regularization parameters λ_1 and λ_2 were estimated using cross-validation. The number of selected landmarks s was 231 for the del.ic.ious data and 278 for the image data set. This was less than the number of sub-problems in both the mlcs and ml-cca approaches, which were also tuned for optimal prediction error.

Table 2: Test set Hamming loss and F1 measure evaluation of the four classification approaches: mlcs, ml-cca, one vs. all, and moplms. The base classifiers in the reduced space were SVM.

	Delicious		Image	
	Ham. loss	F-score	Ham. loss	F-score
mlcs	0.0187	0.3732	0.0047	0.3012
ml-cca	0.0164	0.3822	0.0041	0.3183
one.vs.all	0.0144	0.4512	0.0034	0.3923
moplms	0.0142	0.4522	0.0032	0.4031

Table 2 displays the F1-score and hamming loss that are two standard evaluation metrics for multi-label classification.

$$\text{Hamming loss} = \frac{y^\top \mathbf{1} + \hat{y}^\top \mathbf{1} - 2y^\top \hat{y}}{k}$$
$$\text{F1 score} = \frac{2y^\top \hat{y}}{\sum_{i=1}^{k} y_i + \sum_{i=1} \hat{y}_i}.$$

The landmark selection method performed better in terms of both evaluation metrics. The one-versus-all method was the second best in terms of prediction accuracy, but takes a significantly greater amount of train and test time, compared to the alternative methods.

Figure 26 (left and middle) shows the decay of the Hamming loss as a function of the sample size for mlcs, moplms and ml-cca method. We omitted the one-vs-all method as it took significantly more amount of time compared to the other approaches, and thus is not computationally attractive. The proposed landmark selection approach has lower prediction error than mlcs and ml-cca.

4.6.2.2 Regression

In the regression setting, we consider predicting the stock prices of several companies based on previous values via the landmark selection approach on the SP 500 data set. More specifically, the data consists of closing stock prices of the 500 companies in the S&P index in the period from August 21, 2009 to August 20, 2010 (a total of

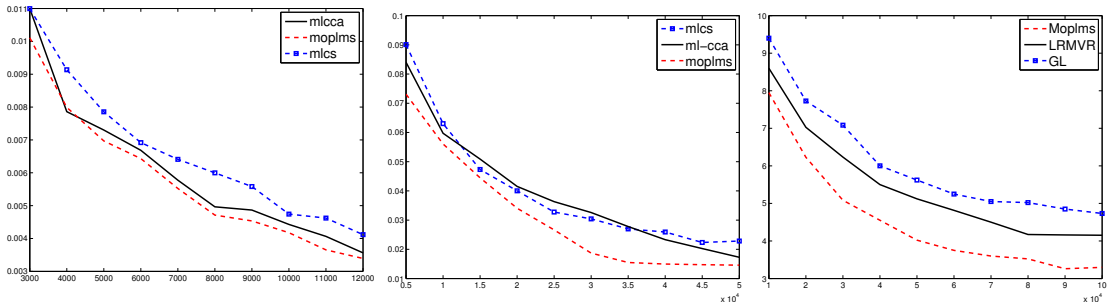


Figure 26: Left and middle: Hamming loss versus number of samples for moplms, mlcs and ml-cca on delicious data set (left) and image data set (middle). Right: Mean MSE prediction error as a function of sample size for moplms, low rank multivariate regression and group Lasso based multivariate regression.

245 entries). We assume the following autoregressive 1 or AR(1) model

$$Y_{tL} = BY_{t-1L} + E \tag{77}$$

where $Y_t = \log \frac{S_t}{S_{t-1}}$ represents the log returns (S_t is the stock price at time t) for day t and E is the noise matrix. The problem is motivated by the observation in finance that multiple companies have stock prices that share identical stochastic trends (cointegration).

We compare our landmark selection approach to low-rank multivariate regression (using trace norm regularization) and group lasso based multivariate regression. These two baselines are popular multivariate regression methods. In our case (moplms), we used a multivariate ridge regression for estimating model (72), which is Equation 77 in the current setting. As in the classification setting, the regularization parameter was tuned by cross validation, and resulted in $s = 98$ landmark outputs.

Table 3 shows that moplms outperformed the two baselines (group lasso and low-rank multivariate regression). Figure 26 displays the prediction error rate as a function of the sample size. It confirms this conclusion as the prediction error of moplms decays faster than the baselines.

Table 3: Test prediction error (MSE) for moplms vs. Lrmvr. λ_1 and λ_2 in state 1 were selected to minimize prediction error using cross-validation. The number of subproblems selected in this case was 98.

Method	Moplms	Group lasso	LRMV Reg
Test err	3.28	5.42	4.63

CHAPTER V

OPTIMAL RANDOM EFFECTS MODEL FOR SPARSE MULTI-TASK LEARNING

5.1 Introduction

Modern high-dimensional data sets, typically with more parameters to estimate than the number of samples available, have triggered a flurry of research based on structured sparse models, both on the statistical and computational aspects. The initial problem considered in this setting was to estimate a sparse vector under a linear model (or the Lasso problem). Recently, several approaches have been proposed for estimating a sparse vector under additional constraints, for e.g., group sparsity- where certain groups of coefficients are jointly zero or non-zero. Another closely related problem is that of multi-task learning or simultaneous sparse approximation, which are special cases of the group sparse formulation. A de-facto procedure for dealing with joint sparsity regularization is the group-Lasso estimator [126], which is based on a $(2,1)$-mixed norm convex relaxation to the non-convex $(2,0)$-mixed norm formulation.

However, as we shall argue in this chapter, group-Lasso suffers from several drawbacks due to the looseness of the relaxation; cf., [57, 47]. We propose a general method for multi-task learning in high-dimensions based on a joint sparsity random effects model. The standard approach for dealing with random effects requires estimating covariance information. Similarly, our estimation procedure involves two-steps: a convex covariance estimation step followed by the standard ridge-regression. The first step corresponds to estimating the covariance of the coefficients under additional constraints that promote sparsity. The intuition is that to deal with group sparsity

(even if we are interested in estimating the coefficients) it is better to first estimate covariance information, and then plug in the covariance estimate for estimating the coefficients. With a particular sparse diagonal structure for the covariance matrix the model becomes similar to group-lasso, and the advantage of the proposed estimation approach over group-lasso formulation will be clarified in this setting.

Related work: Traditional estimation approaches for random effects model involve two-steps: first estimate the underlying covariance matrix, and then estimate the coefficients based on the covariance matrix. However, the traditional covariance estimation procedures are non-convex such as the popular method of restricted maximum likelihood (*REML*) and such models are typically studied in the low-dimensional setting [51].

From a Bayesian perspective, a hierarchical model for simultaneous sparse approximation is proposed in [119] based on a straightforward extension of automatic relevance determination. Under that setting, the tasks share a common hyper-prior that is estimated from the data by integrating out the actual parameter. The resulting marginal likelihood is maximized for the hyper-prior parameters; this procedure is called as type-II maximum likelihood in the literature. The non-Bayesian counterpart is called *random effects model* in classical statistics, and the resulting estimator is referred to as REML. The disadvantage of this approach is that it makes the resulting optimization problem non-convex and difficult to solve efficiently, as mentioned before. In addition, the problem becomes harder to analyze and provide convincing statistical and computational guarantees, while Lasso-related formulations are well studied and favorable statistical and computational properties could be established.

More recently, the problem of joint sparsity regularization has been studied under various settings (multi-task learning [2, 1], group lasso [126], and simultaneous sparse approximation [110, 119]) in the past years. In [1], the authors develop a convex

framework for multi-task learning based on the $(2, 1)$-mixed norm formulation. Conditions for sparsity oracle inequalities and variable selection properties for a similar formulation are derived in [72], showing the advantage of joint estimation of tasks that share common support is statistically efficient. But the formulation has several drawbacks due to the looseness of its convex relaxation [57, 47]. The issue of bias that is inherent in the group lasso formulation was discussed in [57]. By defining a measure of sparsity level of the target signal under the group setting, the authors mention that the standard formulation of group lasso exhibits a bias that cannot be removed by simple reformulation of group lasso. In order to deal with this issue, recently [47] proposed the use of a non-convex regularizer and provided a numerical algorithm based on solving a sequence of convex relaxation problems. The method is based on a straightforward extension of approach developed for the Lasso setting (cf., [128]), to the joint sparsity situation. Note that adaptive group-Lasso is a special case of [47]. In this chapter, we propose a simple two-step procedure, to overcome the drawbacks of the standard group-Lasso relaxation. Compared to [47], the proposed approach is entirely convex and hence attains the global solution.

The current chapter has two theoretical contributions. First, under a multi-task random effects model, we obtain an expected prediction error bound that relates the predictive performance to the accuracy of covariance estimation; by adapting high dimensional sparse covariance estimation procedures such as [36, 15], we can obtain consistent estimate of covariance matrix which leads to asymptotically optimal performance. Second, it is shown that under our random effects model, group Lasso in general does not accurately estimate the covariance matrix and thus is not optimal under the model considered. Experiments show that this approach provides improved performance compared to group Lasso (and the multi-stage versions) on simulated and real data sets.

5.2 Joint Sparsity Random Effects Model and Group Lasso

We consider joint sparsity regularization problems under multi-task learning. In multi-task learning, we consider m linear regression problems tasks $\ell = 1, \ldots, m$

$$Y^{(\ell)} = X^{(\ell)}\bar{\beta}^{(\ell)} + \epsilon^{(\ell)}. \tag{78}$$

We assume that each $Y^{(\ell)}$ is an $n^{(\ell)}$ dimensional vector, each $X^{(\ell)}$ is an $n^{(\ell)} \times d$ dimensional matrix, each $\bar{\beta}^{(\ell)}$ is the target coefficient vector for task ℓ in d dimension. For simplicity, we also assume that $\epsilon^{(\ell)}$ is an $n^{(\ell)}$ dimensional iid zero-mean Gaussian noise vector with variance σ^2: $\epsilon^{(\ell)} \sim N(0, \sigma^2 I_{n^{(\ell)} \times n^{(\ell)}})$.

The joint sparsity model in multi-task learning assumes that all $\bar{\beta}^{(\ell)}$ share similar supports: $\mathrm{supp}(\bar{\beta}^{(\ell)}) \subset \bar{F}$ for some common sparsity pattern $\bar{F}$, where $\mathrm{supp}(\beta) = \{j : \beta_j \neq 0\}$. The convex relaxation formulation for this model is given by group Lasso

$$\min_{\beta} \left[\sum_{\ell=1}^{m} \frac{1}{2} \left\| Y^{(\ell)} - X^{(\ell)}\beta^{(\ell)} \right\|_2^2 + \lambda \sum_{j=1}^{d} \sqrt{\sum_{\ell=1}^{m} (\beta_j^{(\ell)})^2} \right], \tag{79}$$

where $\beta = \{\beta^{(\ell)}\}_{\ell=1,\ldots,m}$.

We observe that the multi-task group Lasso formulation (79) is equivalent to $\min_{\beta,\omega} F(\beta, \omega)$, where $F(\beta, \omega) =$

$$\sum_{\ell=1}^{m} \frac{1}{2\sigma^2} \left\| Y^{(\ell)} - X^{(\ell)}\beta^{(\ell)} \right\|_2^2 + \sum_{j=1}^{d} \frac{1}{2\omega_j} \sum_{\ell=1}^{m} (\beta_j^{(\ell)})^2 + \frac{m}{2\sigma^2} \sum_{j=1}^{d} \omega_j \tag{80}$$

with $\lambda = \sigma\sqrt{m}$, where $\beta = \{\beta^{(\ell)}\}_{\ell=1,\ldots,m}$ and $\omega = \{\omega_j\}_{j=1,\ldots,d}$.

With fixed hyper parameter ω, we note that (79) is a special case of

$$\min_{\beta} \sum_{\ell=1}^{m} \frac{1}{2\sigma^2} \left\| Y^{(\ell)} - X^{(\ell)}\beta^{(\ell)} \right\|_2^2 + \frac{1}{2} \sum_{\ell=1}^{m} (\beta^{(\ell)})^\top \Omega^{-1} \beta^{(\ell)}, \tag{81}$$

where Ω is a hyper parameter covariance matrix shared among the tasks. This general method employs a common quadratic regularizer that is shared by all the tasks. The group Lasso formulation (79) assumes a specific form of diagonal covariance matrix $\Omega = \mathrm{diag}(\{\omega_j\})$.

Equation (81) suggests the following random effects model for joint sparsity regularization, where the coefficient vectors $\bar{\beta}^{(\ell)}$ are random vectors generated independently for each task ℓ; however they share the same covariance matrix $\bar{\Omega}$: $E\ \bar{\beta}^{(\ell)}\bar{\beta}^{(\ell)\top} = \bar{\Omega}$. Given the coefficient vector $\bar{\beta}$, we then generate $Y^{(\ell)}$ based on (81). Note that we assume that Ω may contain zero-diagonal elements. If $\Omega_{jj} = 0$, then the corresponding $\bar{\beta}_j^{(\ell)} = 0$ for all ℓ. Therefore we call this model *joint sparsity random effects model* for multi-task learning.

5.3 Joint Sparsity via Covariance Estimation

Under the proposed joint sparsity random effects model, it can be shown (see Section 5.4) that the optimal quadratic optimizer $(\beta^{(\ell)})^\top \Omega^{-1} \beta^{(\ell)}$ in (79) is obtained at the true covariance $\Omega = \bar{\Omega}$. This observation suggests the following estimation procedure involving two steps:

- Step 1: Estimate the joint covariance matrix Ω as hyper parameter. In particular, this chapter suggests the following method as discussed in Section 5.3.1:
  $$\hat{\Omega} = \arg\min_{\Omega \in \mathcal{S}} \left[\frac{1}{2} \sum_{\ell=1}^{m} \left\| Y^{(\ell)} Y^{(\ell)\top} - X^{(\ell)} \Omega X^{(\ell)\top} \right\|_F^2 + R(\Omega) \right], \tag{82}$$
  where $\|\cdot\|_F$ denotes the matrix Frobenius norm, $\mathcal{S}$ is the set of symmetric positive semi-definite matrices, and $R(\Omega)$ is an appropriately defined regularizer function (specified in Section 5.3.1).

- Step 2: Compute each $\beta^{(\ell)}$ separately given the estimated $\hat{\Omega}$ using:
  $$\hat{\beta}^{(\ell)} = \left(X^{(\ell)\top} X^{(\ell)} + \lambda \hat{\Omega}^{-1} \right)^{-1} X^{(\ell)\top} Y^{(\ell)}, \tag{83}$$
  where $\ell = 1, \ldots, m$.

Note that the estimation method proposed in step 1 holds for a general class of covariance matrices. Meaningful estimates of the covariance matrix could be obtained

even when the generative model assumption is violated. If the dimension d and sample size n per task are fixed, it can be shown relatively easily using classical asymptotic statistics that when $m \to \infty$, we can reliably estimate the true covariance $\bar{\Omega}$ using (82), i.e., $\hat{\Omega} \to \bar{\Omega}$. Therefore the method is asymptotically optimal as $m \to \infty$. On the other hand, the group Lasso formulation (80) produces sub-optimal estimate of ω_j, as we shall see in Section 5.4.2. We would like to point out that in cases when the matrix $\hat{\Omega}$ is not invertible (for example, as in the sparse diagonal case as we see next) we replace the inverse with pseudo-inverse. For ease of presentation, we use the inverse throughout the presentation, though it should be clear from the context.

5.3.1 Sparse Covariance Coding Models

In our two step procedure, the covariance estimation of step 1 is more complex compared to step 2, which involves only the solutions of ridge regression problems. As mentioned above, if we employ a full covariance estimation model, then the estimation procedure proposed in this work is asymptotically optimal when $m \to \infty$. However, since modern asymptotics are often concerned with the scenario when $d \gg n$, computing a $d \times d$ full matrix Ω becomes impossible without further structure on Ω. In this section, we assume that Ω is diagonal, which is consistent with the group Lasso model.

This section explains how to estimate Ω using our generative model, which implies that $\bar{\beta}^{(\ell)} \sim N(0, \Omega)$, and $Y^{(\ell)} = X^{(\ell)}\bar{\beta}^{(\ell)} + \epsilon^{(\ell)}$ with $\epsilon^{(\ell)} \sim N(0, \sigma^2 I_{n^{(\ell)} \times n^{(\ell)}})$. Taking expectation of $Y^{(\ell)}Y^{(\ell)\top}$ with respect to ϵ and $\bar{\beta}^{(\ell)}$, we obtain $E_{\beta^{(\ell)},\epsilon} Y^{(\ell)}Y^{(\ell)\top} = X^{(\ell)}\Omega X^{(\ell)\top} + \sigma^2 I_{n^{(\ell)} \times n^{(\ell)}}$. This suggests the following estimator of Ω: $\hat{\Omega} =$

$$\arg\min_{\Omega \in \mathcal{S}} \sum_{\ell=1}^{m} \left\| Y^{(\ell)}Y^{(\ell)\top} - X^{(\ell)}\Omega X^{(\ell)\top} - \sigma^2 I_{n^{(\ell)} \times n^{(\ell)}} \right\|_F^2,$$

where $\|\cdot\|_F$ is the matrix Frobenius norm. This is equivalent to

$$\hat{\Omega} = \arg\min_{\Omega \in \mathcal{S}} \frac{1}{2} \sum_{\ell=1}^{m} \left\| Y^{(\ell)}Y^{(\ell)\top} - X^{(\ell)}\Omega X^{(\ell)\top} \right\|_F^2 + \lambda \mathrm{tr}\left(\Omega \sum_{\ell=1}^{m} X^{(\ell)\top} X^{(\ell)} \right) \qquad (84)$$

with $\lambda = \sigma^2$. Similar ideas for estimating covariance by this approach appeared in [36, 16]. We may treat the last term as regularizer of Ω, and in such sense a more general form is to consider $\hat{\Omega} =$

$$\arg\min_{\Omega \in \mathcal{S}} \left[\frac{1}{2} \sum_{\ell=1}^{m} \left\| Y^{(\ell)} Y^{(\ell)\top} - X^{(\ell)} \Omega X^{(\ell)\top} \right\|_F^2 + R(\Omega) \right],$$

where $R(\Omega)$ is a general regularizer function of Ω. Note that the dimension d can be large, and thus special structure is needed to regularize Ω. In particular, to be consistent with group Lasso, we impose the diagonal covariance constraint $\Omega = \mathrm{diag}(\{\omega_j\})$, and then encourage sparsity as follows: $\hat{\Omega} =$

$$\arg\min_{\{\omega_j \geq 0\}} \sum_{\ell=1}^{m} \frac{1}{2} \| Y^{(\ell)} Y^{(\ell)\top} - X^{(\ell)} \mathrm{diag}(\{\omega_j\}) X^{(\ell)\top} \|_F^2 + \lambda \sum_j \omega_j. \tag{85}$$

This formulation leads to sparse estimation of ω_j, which we call *sparse covariance coding (scc)*. Note that the above optimization problem is convex and hence the solution could be computed efficiently. This formulation is consistent with the group Lasso regularization which also assumes diagonal covariance implicitly as in (79). It should be noted that if the diagonals of $\sum_{\ell=1}^{m} X^{(\ell)\top} X^{(\ell)}$ have identical values, then up to a rescaling of λ, (85) is equivalent to (84) with Ω restricted to be a diagonal matrix. In the experiments conducted on real world data sets, there was no significant difference between the two regularization terms (see Table 7), when both formulations are restricted to diagonal Ω.

5.4 Theoretical Analysis

In this section we do a theoretical analysis of the proposed method. Specifically, we first derive upper and lower bounds for prediction error for the joint sparsity random effects model and show the optimality of the proposed approach. Informally, the notion of optimality considered is as follows: what is the 'optimal shared quadratic regularizer', when m and d goes to infinity and when solutions for each task can be written as individual ridge regression solutions with a shared quadratic regularizer

(note that this includes group-Lasso method). Next, we demonstrate with a simple example (i.e., considering the low-dimensional setting) the drawback of the standard group-Lasso relaxation. In a way, this example also serves as a motivation for the approach proposed in this work and provides concrete intuition.

We consider a simplified analysis with $\hat{\Omega}$ replaced by $\hat{\Omega}^{(\ell)}$ in Step 2 so that $\hat{\Omega}^{(\ell)}$ does not depend on $Y^{(\ell)}$:

$$\hat{\beta}^{(\ell)} = \left(X^{(\ell)\top} X^{(\ell)} + \lambda \hat{\Omega}^{(\ell)-1}\right)^{-1} X^{(\ell)\top} Y^{(\ell)}. \tag{86}$$

For example, this can be achieved by replacing Step 1 with $\hat{\Omega}^{(\ell)} =$

$$\arg\min_{\Omega \in \mathcal{S}} \left[\frac{1}{2} \sum_{k \neq \ell} \left\| Y^{(k)} Y^{(k)\top} - X^{(k)} \Omega X^{(k)\top} \right\|_F^2 + R(\Omega)\right]. \tag{87}$$

Obviously when m is large, we have $\hat{\Omega}^{(\ell)} \approx \hat{\Omega}$. Therefore the analysis can be slightly modified to the original formulation, with an extra error term of $O(1/m)$ that vanishes when $m \to \infty$. Nevertheless, the independence of $\hat{\Omega}^{(\ell)}$ and $Y^{(\ell)}$ simplifies the argument and makes the essence of our analysis much easier to understand.

5.4.1 Prediction Error

This section derives an expected prediction error bound for the coefficient vector $\hat{\beta}^{(\ell)}$ in (86) in terms of the accuracy of the covariance matrix estimation $\hat{\Omega}^{(\ell)}$. We consider the fixed design scenario, where the design matrices $X^{(\ell)}$ are fixed and $\epsilon^{(\ell)}$ and $\bar{\beta}^{(\ell)}$ are random.

Theorem 5.4.1. *Assume that $\lambda \geq \sigma^2$. For each task ℓ, given $\hat{\Omega}^{(\ell)}$ that is independent of $Y^{(\ell)}$, the expected prediction error with $\hat{\beta}^{(\ell)}$ in (86) is bounded as*

$$\sigma^2 \lambda \omega^{(\ell)} \leq A \leq \lambda^2 \omega^{(\ell)},$$

where $A = E\left\|X^{(\ell)}\hat{\beta}^{(\ell)} - X^{(\ell)}\bar{\beta}^{(\ell)}\right\|_2^2 - \left\|X^{(\ell)}\bar{\Omega}^{1/2}\left(\bar{\Omega}^{1/2}\Sigma^{(\ell)}\bar{\Omega}^{1/2} + \lambda I\right)^{-1/2}\right\|_F^2$ and the expectation is with respect to the random effects $\bar{\beta}^{(\ell)}$ and noise $\epsilon^{(\ell)}$, and $\Sigma^{(\ell)} = X^{(\ell)\top} X^{(\ell)}$,

and

$$\omega^{(\ell)} = \|X^{(\ell)}\left(\hat{\Omega}^{(\ell)}\Sigma^{(\ell)} + \lambda I\right)^{-1}(\hat{\Omega}^{(\ell)} - \bar{\Omega})(\Sigma^{(\ell)})^{1/2}\left((\Sigma^{(\ell)})^{1/2}\bar{\Omega}(\Sigma^{(\ell)})^{1/2} + \lambda I\right)^{-1/2}\|_F^2.$$

Proof. For notational simplicity, we remove the superscripts (ℓ) in the following derivation (e.g., denote $X^{(\ell)}$ by X, $\hat{\beta}^{(\ell)}$ by $\hat{\beta}$ and so on). We have the following decomposition

$$\begin{aligned}
&E\,\|X\hat{\beta} - X\bar{\beta}\|_2^2 \\
=&E\,\left\|X\left(\left(X^\top X + \lambda\hat{\Omega}^{-1}\right)^{-1}X^\top(X\bar{\beta} + \epsilon) - \bar{\beta}\right)\right\|_2^2 \\
=&E\,\left\|X\left(X^\top X + \lambda\hat{\Omega}^{-1}\right)^{-1}\lambda\hat{\Omega}^{-1}\bar{\beta}\right\|_2^2 + E\,\left\|X\left(X^\top X + \lambda\hat{\Omega}^{-1}\right)^{-1}X^\top\epsilon\right\|_2^2 \\
=&\lambda^2\mathrm{tr}\left[X\left(X^\top X + \lambda\hat{\Omega}^{-1}\right)^{-1}\hat{\Omega}^{-1}\bar{\Omega}\hat{\Omega}^{-1}\left(X^\top X + \lambda\hat{\Omega}^{-1}\right)^{-1}X^\top\right] \\
&+ \sigma^2\mathrm{tr}\left[X\left(X^\top X + \lambda\hat{\Omega}^{-1}\right)^{-1}X^\top X\left(X^\top X + \lambda\hat{\Omega}^{-1}\right)^{-1}X^\top\right] \\
\leq&\mathrm{tr}\lambda\left[X\left(\hat{\Omega}X^\top X + \lambda I\right)^{-1}(\lambda\bar{\Omega} + \hat{\Omega}X^\top X\hat{\Omega})\left(X^\top X\hat{\Omega} + \lambda I\right)^{-1}X^\top\right] \\
=&\lambda(A + B + C),
\end{aligned}$$

where with $\Delta\hat{\Omega} = \hat{\Omega} - \bar{\Omega}$, we have

$$A = \mathrm{tr}\left[X\left(\hat{\Omega}X^\top X + \lambda I\right)^{-1}\Delta\hat{\Omega}X^\top X\Delta\hat{\Omega}\left(X^\top X\hat{\Omega} + \lambda I\right)^{-1}X^\top\right]$$

and

$$B = 2\mathrm{tr}\left[X\left(\hat{\Omega}X^\top X + \lambda I\right)^{-1}\bar{\Omega}X^\top X\Delta\hat{\Omega}\left(X^\top X\hat{\Omega} + \lambda I\right)^{-1}X^\top\right]$$

and

$$C = \mathrm{tr}\left[X\left(\hat{\Omega}X^\top X + \lambda I\right)^{-1}(\bar{\Omega}X^\top X\bar{\Omega} + \lambda\bar{\Omega})\left(X^\top X\hat{\Omega} + \lambda I\right)^{-1}X^\top\right].$$

We can further expand C as:

$$\begin{aligned}
C =& \text{tr}\left[X\left(\bar{\Omega}X^\top X+\lambda I\right)^{-1}\left(\bar{\Omega}X^\top X\bar{\Omega}+\lambda\bar{\Omega}\right)\left(X^\top X\hat{\Omega}+\lambda I\right)^{-1}X^\top\right] \\
&-\text{tr}\left[X\left(\hat{\Omega}X^\top X+\lambda I\right)^{-1}\Delta\hat{\Omega}X^\top X\left(\bar{\Omega}X^\top X+\lambda I\right)^{-1}\left(\bar{\Omega}X^\top X\bar{\Omega}+\lambda\bar{\Omega}\right)\left(X^\top X\hat{\Omega}+\lambda I\right)^{-1}X^\top\right] \\
=& \text{tr}\left[X\bar{\Omega}\left(X^\top X\hat{\Omega}+\lambda I\right)^{-1}X^\top\right]-\text{tr}\left[X\left(\hat{\Omega}X^\top X+\lambda I\right)^{-1}\Delta\hat{\Omega}X^\top X\bar{\Omega}\left(X^\top X\hat{\Omega}+\lambda I\right)^{-1}X^\top\right] \\
=& \text{tr}\left[X\bar{\Omega}\left(X^\top X\hat{\Omega}+\lambda I\right)^{-1}X^\top\right]-B/2.
\end{aligned}$$

Therefore

$$\begin{aligned}
& B+C-\text{tr}\left[X\bar{\Omega}\left(X^\top X\bar{\Omega}+\lambda I\right)^{-1}X^\top\right] \\
=& B/2-\text{tr}\left[X\bar{\Omega}\left(X^\top X\bar{\Omega}+\lambda I\right)^{-1}X^\top X\Delta\hat{\Omega}\left(X^\top X\hat{\Omega}+\lambda I\right)^{-1}X^\top\right] \\
=& B/2-\text{tr}\left[X\left(\bar{\Omega}X^\top X+\lambda I\right)^{-1}\bar{\Omega}X^\top X\Delta\hat{\Omega}\left(X^\top X\hat{\Omega}+\lambda I\right)^{-1}X^\top\right] \\
=& -\text{tr}\left[X\left(\hat{\Omega}X^\top X+\lambda I\right)^{-1}\Delta\hat{\Omega}X^\top X\left(\bar{\Omega}X^\top X+\lambda I\right)^{-1}\bar{\Omega}X^\top X\Delta\hat{\Omega}\left(X^\top X\hat{\Omega}+\lambda I\right)^{-1}X^\top\right].
\end{aligned}$$

Therefore we have

$$\begin{aligned}
& A+B+C-\text{tr}\left[X\bar{\Omega}\left(X^\top X\bar{\Omega}+\lambda I\right)^{-1}X^\top\right] \\
=& \text{tr}\left[X\left(\hat{\Omega}X^\top X+\lambda I\right)^{-1}\Delta\hat{\Omega}\left(I-X^\top X\left(\bar{\Omega}X^\top X+\lambda I\right)^{-1}\bar{\Omega}\right)X^\top X\Delta\hat{\Omega}\left(X^\top X\hat{\Omega}+\lambda I\right)^{-1}X^\top\right] \\
=& \lambda\text{tr}\left[X\left(\hat{\Omega}X^\top X+\lambda I\right)^{-1}\Delta\hat{\Omega}\left(X^\top X\bar{\Omega}+\lambda I\right)^{-1}X^\top X\Delta\hat{\Omega}\left(X^\top X\hat{\Omega}+\lambda I\right)^{-1}X^\top\right].
\end{aligned}$$

This proves the upper bound. Similarly, the lower bound follows from the fact that $E\left\|X\hat{\beta}-X\bar{\beta}\right\|_2^2\geq\sigma^2(A+B+C)$.

□

The bound shows that the prediction performance of (86) depends on the accuracy of estimating $\bar{\Omega}$. In particular, if $\hat{\Omega}^{(\ell)}=\bar{\Omega}$, then the optimal prediction error of $\left\|X^{(\ell)}\bar{\Omega}^{1/2}\left(\bar{\Omega}^{1/2}X^{(\ell)\top}X^{(\ell)}\bar{\Omega}^{1/2}+\lambda I\right)^{-1/2}\right\|_F^2$ can be achieved. A simplified upper bound is $E\left\|X^{(\ell)}\hat{\beta}^{(\ell)}-X^{(\ell)}\bar{\beta}^{(\ell)}\right\|_2^2\leq\left\|X^{(\ell)}\bar{\Omega}^{1/2}\left(\bar{\Omega}^{1/2}\Sigma^{(\ell)}\bar{\Omega}^{1/2}+\lambda I\right)^{-\frac{1}{2}}\right\|_F^2+\lambda^{-1}\|\Sigma^{(\ell)}(\hat{\Omega}^{(\ell)}-\bar{\Omega})\|_F^2$.

This means that if the covariance estimation is consistent; that is, if $\hat{\Omega}^{(\ell)}$ converges to $\bar{\Omega}$, then our method achieves the optimal prediction error $\left\| X^{(\ell)} \bar{\Omega}^{1/2} \left(\bar{\Omega}^{1/2} \Sigma^{(\ell)} \bar{\Omega}^{1/2} + \lambda I \right)^{-1/2} \right\|_F^2$ for all tasks.

The consistency of $\hat{\Omega}^{(\ell)}$ has been studied in the literature, for example by [15] under high dimensional sparsity assumptions. Such results can be immediately applied with Theorem 6.5.1 to obtain optimality of the proposed approach. Specifically, we consider the case of diagonal covariance matrix, where the sparsity in $\bar{\Omega}$ is defined as the number of non-zero diagonal entries, i.e., $s = |\{i : \Omega_{ii} \neq 0\}|$. Following [15], we consider the case $X^{(\ell)} = X \in \mathbb{R}^{n \times d}, \ell = 1, \ldots, m$. Let X_J denote the sub matrix of X obtained by removing the columns of X whose indices are not in the set J. We also assume that the diagonals of $X^\top X$ have identical values so that (85) is equivalent to (84) up to a scaling of λ.

Let $\rho_{\min}(A)$ and $\rho_{\max}(A)$ for a matrix A denote the smallest and largest eigenvalue of A respectively. We introduce two quantities [15] that impose certain assumptions on the matrix X.

Definition 4. For $0 < t \leq d$, define $\rho_{\min}(t) := \inf_{\substack{J \subset \{1,\ldots,d\} \\ |J| \leq t}} \rho_{\min}(X_J^\top X_J)$.

Definition 5. The mutual coherence of the columns $X_t, t = 1, \ldots, d$ of X is defined as $\theta(X) := \max\{|X_{t'}^\top X_t|, t \neq s', 1 \leq t, t' \leq d\}$ and let $X_{\max}^2 := \max\{\|X_t\|_2^2, 1 \leq t \leq d\}$.

We now state the following theorem establishing the consistency of covariance estimation (given by Eq 87) in the high-dimensional setting. The proof essentially follows the same argument for Theorem 8 in [15], by noticing the equivalence between (85) and (84), which implies consistency.

Theorem 5.4.2. *Assume that $\bar{\Omega}$ is diagonal, and $\theta(X) < \rho_{\min}(s)^2 / 4\rho_{\max}(X^\top X)s$. Assume n is fixed and the number of tasks and dimensionality $m, d \to \infty$ such that $\sqrt{s} \ln d / m \to 0$. Then the covariance estimator of (87), with appropriately chosen λ*

and $R(\Omega)$ defined by (85), converges to $\bar{\Omega}$:

$$\|X(\hat{\Omega}^{(\ell)} - \bar{\Omega})X^\top\|_F^2 \to_P 0. \tag{88}$$

The following corollary, which is an immediate consequence of Theorem 6.5.1 and 5.4.2, establishes the asymptotic optimality (for prediction) of the proposed approach under the sparse diagonal matrix setting and $R(\Omega)$ defined as in (85). Similar result could be derived for other regularizers for $R(\Omega)$.

Corollary 5. *Under the assumption of Theorem 6.5.1 and 5.4.2, the two-step approach defined by (87) and (86), with $R(\Omega)$ defined by (85) is asymptotically optimal for prediction, for each task ℓ:*

$$E\,\|X\hat{\beta}^{(\ell)} - X\bar{\beta}^{(\ell)}\|_2^2 - \left\|X\bar{\Omega}^{1/2}\left(\bar{\Omega}^{1/2}X^\top X\bar{\Omega}^{1/2} + \lambda I\right)^{-1/2}\right\|_F^2 \to_P 0.$$

Note that the asymptotics considered above, reveals the advantage of *multi-task learning* under the joint sparsity assumption: with a fixed number of samples per each task, as the dimensions of the samples and *number of tasks* tend to infinity (obeying the condition given in theorem 5.4.2) the proposed two-step procedure is asymptotically optimal for prediction. Although for simplicity, we state the optimality result for (87) and (86), the same result holds for the two-step procedure given by (82) and (83), because $\hat{\Omega}^{(\ell)}$ of (87) and $\hat{\Omega}$ of (82) differ only by a factor of $O(1/m)$ which converges to zero under the asymptotics considered. Finally, we would like to remark that the mutual coherence assumption made in Theorem 5.4.2 could be relaxed to milder conditions (based on restricted eigenvalue type assumptions) - we leave it as future work.

5.4.2 Drawback of Group Lasso

In general, group Lasso does not lead to optimal performance due to looseness of the single step convex relaxation. [57, 47]. This section presents a simple but concrete example to illustrate the phenomenon and shows how $\bar{\Omega}$ is under-estimated in the

group-Lasso formulation. Combined with the previous section, we have a complete theoretical justification of the superiority of our approach over group Lasso, which we will also demonstrate in the empirical study.

For this purpose, we only need to consider the following relatively simple illustration (in the low-dimensional setting). We consider the case when all design matrices equal identity: $X^{(\ell)} = I$ for $\ell = 1, \ldots, m$. This formulation is similar to *Normal means models*, a popular model in the statistics literature. It is instructive to consider this model because of its closed form solution. It helps in deriving useful insights that further help for a better understanding of more general cases. We are interested in the asymptotic behavior when $m \to \infty$ (with $n^{(\ell)}$ and d fixed), which simplifies the analysis, but nevertheless reveals the problems associated with the standard group Lasso formulation. Moreover, it should be mentioned that although the two-step procedure is motivated from a generative model, the analysis presented in this section does not need to assume that each $\beta^{(\ell)}$ is truly generated from such a model.

Proposition 16. *Suppose that $n^{(\ell)} = d$ and $X^{(\ell)} = I$ for $\ell = 1, \ldots, m$, and $m \to \infty$. The sparse covariance estimate corresponding to the formulation defined by (85) is consistent.*

Proof. The sparse covariance coding formulation (85) is equivalent to (with the intention of setting $\lambda = \sigma^2$): $\hat{\Omega}^{scc} = \arg\min_{\{\omega_j \geq 0\}} \sum_{\ell=1}^m \frac{1}{2} \left\| Y^{(\ell)} Y^{(\ell)\top} - \mathrm{diag}(\{\omega_j\}) \right\|_F^2 + \lambda m \sum_j \omega_j$. The closed form solution is given by $\hat{\omega}_j^{scc} = \max\left(0, m^{-1} \sum_{\ell=1}^m (Y_j^{(\ell)})^2 - \lambda\right)$ for $j = 1, \ldots, d$. Since $m^{-1} \sum_{\ell=1}^m (Y_j^{(\ell)})^2 \to E_{\beta^{(\ell)}}(\beta_j^{(\ell)})^2 + \sigma^2$ as $m \to \infty$, the variance $\hat{\omega}_j^{scc} \to E_{\beta^{(\ell)}}(\beta_j^{(\ell)})^2$ with $\lambda = \sigma^2$. Therefore $\hat{\omega}_j$ is consistent. □

Note that by plugging-in the estimate of variance into (83) with the same λ (with $\lambda = \sigma^2$), we obtain

$$\hat{\beta}_j^{(\ell)} = Y_j^{(\ell)} \max\left(0, 1 - \frac{\lambda}{m^{-1} \sum_{\ell=1}^m (Y_j^{(\ell)})^2}\right). \tag{89}$$

An immediate consequence of Proposition 16 is that the estimate define in (89) is asymptotically optimal for any method using a quadratic regularizer shared by all the tasks.

A similar analysis of group Lasso formulation would reveal its drawback. Consider the group Lasso formulation defined in (80). Under similar settings, the formulation can be written as $[\hat{\beta}, \hat{\omega}^{gl}] =$

$$\arg\min_{\beta,\omega} \sum_{\ell=1}^{m} \left\| Y^{(\ell)} - \beta^{(\ell)} \right\|_2^2 + \lambda \sum_{j=1}^{d} \frac{1}{\omega_j} \sum_{\ell=1}^{m} (\beta_j^{(\ell)})^2 + m \sum_{j=1}^{d} \omega_j.$$

The closed form solution for the above formulation is given by

$$\hat{\omega}_j^{gl} = \max\left(0, \sqrt{\lambda m^{-1} \sum_{\ell=1}^{m} (Y_j^{(\ell)})^2} - \lambda\right),$$

for $j = 1, \ldots, d$ and the corresponding coefficient estimate is

$$\hat{\beta}_j^{(\ell)} = Y_j^{(\ell)} \max\left(0, 1 - \frac{\sqrt{\lambda}}{\sqrt{m^{-1} \sum_{\ell=1}^{m} (Y_j^{(\ell)})^2}}\right)$$

, for $\ell = 1, \ldots, m$ and $j = 1, \ldots, d$.

The solution for $\hat{\omega}_j^{gl}$ implies that it is not possible to pick a fixed λ such that the group Lasso formulation gives consistent estimate of ω_j. Since from (80), it is evident that group Lasso can also be regarded as a method that uses a quadratic regularizer shared by all the tasks, we know that the solution obtained for the corresponding co-efficient estimate is asymptotically sub-optimal. In fact, the covariance estimate $\hat{\omega}_j^{gl}$ is significantly smaller than the correct estimate $\hat{\omega}_j^{scc}$. This under-estimate of ω_j in group Lasso implies a corresponding under-estimate of $\beta^{(\ell)}$ obtained via group Lasso, when compared to (89). This under-estimation is the underlying theoretical reason why the proposed two-step procedure is superior to group Lasso for learning with joint sparsity. This claim is also confirmed by our empirical studies.

5.4.3 Other Covariance Coding Models

We now demonstrate the generality of the proposed approach for multi-task learning. Note that in addition to the sparse covariance coding method (85) that assumes a diagonal form of Ω plus sparsity constraint, some other structures may be explored. One method that has been suggested for covariance estimation in [15] is the following formulation:

$$\hat{\Omega} = \arg\min_{\Omega \in \mathcal{S}} \sum_{\ell=1}^{m} \|Y^{(\ell)} Y^{(\ell)\top} - X^{(\ell)} \Omega X^{(\ell)}\|_F^2 + 2\lambda \sum_k \gamma_k \sqrt{\sum_m \Omega_{k,m}^2}, \qquad (90)$$

where $\mathcal{S}$ denotes the set of symmetric positive semi-definite matrices $\mathcal{S}$. This approach selects a set of features, and then models a full covariance matrix within the selected set of features. Although the feature selection is achieved with a group Lasso penalty, unlike this work, [15] didn't study the possibility of using covariance estimation to do joint feature selection (which is the main purpose of this work), but rather studied covariance estimation as a separate problem.

The partial full covariance model in (90) has complexity in between that of the full covariance model and the sparse diagonal covariance model (sparse covariance coding) which we promote in this chapter, at least for the purpose of joint feature selection. The latter has the smallest complexity, and thus more effective for high dimensional problems that tend to cause over-fitting.

Another model with complexity in between of sparse diagonal covariance and full covariance model is to model the covariance matrix Ω as the sum of a sparse diagonal component plus a low-rank component. This is similar in spirit to the more general sparse+low-rank matrix decomposition formulation recently appeared in the literature [27, 23, 55]. However since the sparse matrix is diagonal, identifiability holds trivially (as described in the appendix) and hence one could in principal, recover both the diagonal and the low-rank objects individually which preserves the advantages of the diagonal formulation and the richness of low-rank formulation. The model

assumption is $\Omega = \Omega_S + \Omega_L$, where Ω_S is the diagonal matrix and Ω_L is the low-rank matrix. The estimation procedure now becomes the following optimization problem (and the rest follows)

$$[\hat{\Omega}_S, \hat{\Omega}_L] = \underset{\Omega_S, \Omega_L}{\arg\min} \sum_{\ell=1}^{m} \frac{1}{2} \|Y^{(\ell)} Y^{(\ell)\top} - X^{(\ell)} (\Omega_S + \Omega_L) X^{(\ell)^\top}\|_F^2 + \lambda_1 \|\Omega_S\|_{\text{vec}(1)} + \lambda_2 \|\Omega_L\|_*,$$

subject to the condition that Ω_S is a non-negative diagonal matrix, and $\Omega_L \in \mathcal{S}$, where $\|\cdot\|_{\text{vec}(1)}$ is the element-wise $L1$ norm and $\|\cdot\|_*$ corresponds to trace-norm.

5.5 Identifiability of additive structure

The issue of identifiability (which is necessary subsequently for consistency and recovery guarantees) arises when we deal with additive decomposition of the covariance matrix. Here, we discuss about the conditions under which the model is identifiable, i.e., there exist an unique decomposition of the covariance matrix as the summation of the sparse diagonal matrix and low-rank matrix. We follow the discussion used in [55]. Let $\Omega = \Omega_s + \Omega_L$ denote the decomposition where Ω_s denotes the sparse diagonal matrix and Ω_L a low-rank matrix. Intuitively, identifiability holds if the sparse matrix is not low-rank (i.e., the support is sufficiently spread out) and the low-rank matrix is not too sparse (i.e., the singular vectors are away from co-ordinate axis). A formal argument is made based on the above intuition. We defined the following quantities (following [55]) below that measures the non-zero entries in any row or column of Ω_s and sparseness of the singular vectors of Ω_L:

$$\alpha = \max\{\|\text{sign}(\Omega_s)\|_{1\to 1}, \|\text{sign}(\Omega_s)\|_{\infty\to\infty}\}$$

and

$$\beta = \|UU^T\|_\infty + \|VV^T\|_\infty + \|U\|_{2\to\infty}\|V\|_{2\to\infty},$$

where $U, V \in \mathbb{R}^{d\times r}$ are the left and right orthonormal singular vectors corresponding to non-zero singular values of Ω_L and $\|M\|_{p\to q} \stackrel{\text{def}}{=} \{\|Mv\|_q : v \in \mathcal{R}^m, \|v\|_p \leq 1\}$.

Note that, for a diagonal matrix, $\|\text{sign}(\Omega_s)\|_{1\to 1} = \|\text{sign}(\Omega_s)\|_{\infty\to\infty} = 1$. It is proved in [55] that if $\alpha\beta < 1$, then the matrices are identifiable, i.e, the sparse plus low-rank decomposition is unique. Therefore we only need to require $\beta < 1$ for identifiability, which is a rather weak assumption, satisfied by most low-rank matrices with sufficient spread of the support.

5.6 Experiments

We demonstrate the advantage of the proposed two-step procedure through (i) multi-task learning experiments on synthetic and real-world data sets and (ii) sparse co-variance coding based image classification.

5.6.1 Multi-task learning

We first report illustrative experiments conducted on synthetic data sets with the proposed models. They are compared with the standard group-lasso formulation. The experimental set up is as follows: the number of tasks $m = 30$, $d = 256$, and $n^\ell = 150$. The data matrix consists of entries from standard Gaussian $N(0, 1)$. To generate the sparse co-efficients, we first generate a random Gaussian vector in d dimensions and set to zero $d - k$ of the co-efficients to account for sparsity. The cardinality of the set of non-zero coefficients is varied as $k = 50, 70, 90$ and the noise variance was 0.1. The results reported are averages over 100 random runs. We compare against standard group lasso, MSMTFL [47] (note that this is a non-convex approach, solved by sequence of convex relaxations) and another natural procedure (GLS-LS) where one uses group lasso for feature selection and with the selected features, one does least squares regression to estimate the coefficients. A precise theoretical comparison to MSMTFL procedure is left as future work.

Tables 5 shows the coefficient estimation error when the samples are such that they share 80% as common basis (and the rest 20% is selected randomly from the remaining basis) and when the samples share the same indices of non-zero coefficients

Table 4: Support selection: Hamming distance between true non-zero indices and estimated non-zero indices by the indicated method for all signals.

Method	80% shared basis			Completely shared basis		
	k=50	k=70	k=90	k=50	k=70	k=90
Standard group lasso	0.18	0.22	0.27	0.11	0.16	0.22
MSMTFL	0.15	0.18	0.20	0.07	0.08	0.17
Partial full covariance	0.17	0.20	0.23	0.07	0.11	0.16
Sparse diagonal covariance	0.13	0.16	0.20	0.05	0.09	0.14

(and the actual values vary for each signals). We note that in both cases, the model with diagonal covariance assumption and partial full covariance (Equation 90) outperforms the standard group lasso formulation, with the diagonal assumption performing better because of good estimates. The diagonal+low-rank formulation slightly outperforms the other models as it preserves the advantages of the diagonal model, while at the same time allows for additional modeling capability through the low-rank part, through proper selection of regularization parameters by cross-validation.

Support selection: While the above experiment sheds light on co-efficient estimation error, we performed another experiment to examine the selection properties of the proposed approach. Table 4 shows the hamming distance between selected basis and the actual basis using the different models. Note that Hamming distance is a desired metric for practical applications where exact recovery of the support set is not possible due to low signal-to-noise ratio. The indices with non-zero entry along the diagonal in the model with diagonal covariance assumption correspond to the selected basis. Similarly, indices with non-zero columns (or rows by symmetry) correspond to the selected basis in the partial full covariance model. The advantage of the diagonal assumption for joint feature selection is clearly seen from the table. This superiority in the feature selection process also explains the better performance achieved for co-efficient estimation. A rigorous theoretical study of the feature selection properties is left as future work.

Correlated data: We next study the effect of correlated data set on the proposed

Table 5: Coefficient estimation: Normalized L_2 distance between true coefficients and estimated coefficients by the indicated method. First 5 rows correspond to 80% shared basis and the last 5 rows correspond to fully shared basis.

Method	k=50	k=70	k=90
standard group Lasso	0.1541 ± 0.0045	0.1919 ± 0.0092	0.2404 ± 0.0124
GLS-LS	0.1498 ± 0.0032	0.1901 ± 0.0034	0.2383 ± 0.0342
Partial full covariance	0.1239 ± 0.0063	0.1542 ± 0.0131	0.1992 ± 0.0143
Sparse Diagonal covariance	0.1022 ± 0.0054	0.1393 ± 0.0088	0.1701 ± 0.0104
MSMTFL	0.1276 ± 0.0075	0.1564 ± 0.0153	0.1987 ± 0.0201
Diag+Low-rank covariance	0.1031 ± 0.0042	0.1212 ± 0.0122	0.1532 ± 0.0173
Standard group Lasso	0.1032 ± 0.0086	0.1574 ± 0.0151	0.1733 ± 0.0190
GLS-LS	0.1010 ± 0.0045	0.1532 ± 0.0134	0.1698 ± 0.0430
Partial full covariance	0.0735 ± 0.0078	0.1131 ± 0.0148	0.1576 ± 0.0201
Sparse Diagonal covariance	0.0447 ± 0.0071	0.0828 ± 0.0165	0.1184 ± 0.0198
MSMTFL	0.0643 ± 0.0093	0.0832 ± 0.0200	0.1457 ± 0.0223
Diag+low-rank Covariance	0.0452 ± 0.0084	0.0786 ± 0.0136	0.1012 ± 0.0161

Table 6: Coefficient estimation: Normalized L_2 distance between true coefficients and estimated coefficients by the indicated method with correlated input data.

Method	k=50	k=70	k=90
Group Lasso	0.2012 ± 0.0033	0.2655 ± 0.0132	0.3252 ± 0.0323
GLS-LS	0.2090 ± 0.0098	0.2702 ± 0.0042	0.3304 ± 0.0333
Partial full covariance	0.1706 ± 0.0064	0.2376 ± 0.0224	0.2701 ± 0.0323
Sparse diagonal covariance	0.1634 ± 0.0022	0.2112 ± 0.0073	0.2601 ± 0.0231
MSMTFL	0.1786 ± 0.0023	0.2323 ± 0.0434	0.2776 ± 0.0223
Diag+Low-rank covariance	0.1531 ± 0.0042	0.2002 ± 0.0236	0.2544 ± 0.0145

approach. We generated correlated Gaussian random variables (corresponding to the size of the data matrix) in order to fill the matrix X for each task. The correlation coefficient was fixed at 0.5. We worked with fully overlapped support set. Other problem parameters were retained. We compared the estimation accuracy of the proposed approach with different settings with group lasso and its variants. The results are summarized in Table 6. Note that the proposed approach performs much better than the group-Lasso based counterparts. Precisely characterizing this improvement theoretically would be interesting.

Next, the proposed approach was tested on three standard multi-task regression

Table 7: Multi-task learning: Average (across task) MSE error on the test data set.

Data set	Group lasso	MSMTFL	Sparse diag. Cov.	Corr. Sparse diag (Eq.84)
Computer	1.542 ± 0.043	1.334 ± 0.031	1.223 ± 0.033	1.209 ± 0.054
School	2.202 ± 0.038	2.033 ± 0.241	1.987 ± 0.040	2.012 ± 0.073
Sarcos	9.221 ± 0.051	9.113 ± 0.145	8.983 ± 0.043	9.002 ± 0.032

data sets (computer, school and sarcos data sets) and compared with the standard approach for multi-task learning: mixed $(2, 1)$-norms or group lasso (79). A description of the data sets is given below:

Computer data set: This dataset consists of a survey among 180 people (corresponding to tasks). Each rated the likelihood of purchasing one of 20 different computers. The input consists 13 different computer characteristics, while the output corresponds to ratings. Following [1], we used the first 8 examples per task for training and the last 4 examples per task for testing.

School data set: This dataset is from the London Education Authority and consists of the exam scores of 15362 students from 139 schools (corresponding to tasks). The input consists 4 school-based and 3 student-based attributes, along with the year. The categorical features are replaced with binary features. We use 75% of the data set for training and the rest for testing.

Sarcos data set: The dataset has 44,484 train samples and 4449 test samples. The task is to map a 21-dimensional input space (corresponding to characteristics of robotic arm) to the the output corresponding to seven torque measurement (tasks) to predict the inverse dynamics.

We report the average (across tasks) root mean square error on the test data set in Table 7. Note that the proposed two-step approach performs better than the group lasso approach on all the data sets. The data sets correspond to cases with varied data size and number of tasks. Observe that even with a small training data (computer data set), performance of both our approach is better than the group-lasso approach.

5.6.2 SCC based Image Classification

In this section, we present a novel application of the proposed approach for obtaining sparse codes for gender recognition in CMU Multi-pie data set. The database contains 337 subjects (235 male and 102 female) across simultaneous variations in pose, expression, and illumination. The advantages of jointly coding the extracted local descriptors of an image with respect to a given dictionary for the purpose of classification has been highlighted in [10]. They propose a method based on mixed $(2, 1)$-norm to jointly find a sparse representation of an image based on local descriptors of that image. Following a similar experimental setup, we use the proposed sparse covariance coding approach for attaining the same goal.

Each image is of size 30×40, size of patches is 8×8, and number of overlapping patches per image is 64. Local descriptors for each images are extracted in the form of overlapping patches and a dictionary is learned based on the obtained patches by sparse coding. With the learnt dictionary, the local descriptors of each image is jointly sparse coded via the diagonal covariance matrix assumption and the codes thus obtained are used or classification. This approach is compared with the group sparse coding based approach. Linear SVM is used in the final step for classification. Note that the purpose of the experiment is not learning a dictionary. Table 8 shows the test set and train set error for the classifier thus obtained. Note that the proposed sparse covariance coding based approach outperforms the group sparse coding based approach for gender classification due to its better quality estimates.

Table 8: Face image classification based on gender: Test and Train set error rates for sparse covariance coding and group sparse coding (both with a fixed dictionary).

	Group sparse coding	Sparse cov. coding
Train error	$6.67 \pm 1.34\%$	$5.56 \pm 1.62\%$
Test error	$7.48 \pm 1.54\%$	$6.32 \pm 1.12\%$

5.6.3 Landmark selection

Landmark selection corresponds to the problem of identifying the data samples, which best approximate the given data set. It is useful for image processing and computer vision applications [94]. We used the proposed approach for selecting the landmark points from a given data set and compared it with group-lasso formulation, which could also be used for the same problem. The data set used is CMU multi-pie data set. The experiment is as follows: A set of 5000 images is split as train set (4000) and test set (1000). The images are such that they are captured with different illuminations and viewpoints. There are 249 subjects in total. The goal is find a small set of images from the train set such that the reconstruction error on the train set and hence on the test set is low. Initially, the train data set is split into two unequal parts (3500 and 500). The larger part is used as basis and regressed against the smaller part to get the landmark images and the reconstruction error is calculated for the training set. Given a new point from test set, it is expressed as the linear combination of the selected basis. The coefficients are estimated by least square method and the average reconstruction error is calculated for the entire set of test points.

Table 9 shows the reconstruction error on the test and train data set, where the regularization parameter was calculated by cross-validation. We note that the proposed approach performs better than the group-lasso formulation. The superiority of the proposed approach for joint feature selection, coupled with better coefficient estimation is exploited in this situation to perform joint landmark selection to approximate the given data set.

5.7 A joint framework for covariance and regression coefficient estimation

In this section, we outline a joint framework for estimating both the covariance matrix and regression coefficients simultaneously. In order to describe the procedure, first

Table 9: Simultaneous basis selection for data approximation: Average reconstruction error

Method	No noise	
	Train error	Test error
Group Lasso	$15.35 \pm 0.99\%$	$22.45 \pm 1.93\%$
Diag Co-var	$10.74 \pm 1.03\%$	$15.87 \pm 1.45\%$
Corr. Diag (Eq. 84)	$10.83 \pm 1.26\%$	$15.71 \pm 1.63\%$
	Noise = 0.02	
	Train error	Test error
Group Lasso	$19.32 \pm 1.24\%$	$26.54 \pm 1.86\%$
Diag Co-var	$13.63 \pm 2.25\%$	$18.02 \pm 2.56\%$
Corr. Diag (Eq. 84)	$13.78 \pm 2.02\%$	$17.63 \pm 2.21\%$

we observe that the equivalent multi-task group Lasso formulation, given by

$$\min_{\beta,\omega} \left[\sum_{\ell=1}^{m} \frac{1}{2\sigma^2} \left\| Y^{(\ell)} - X^{(\ell)}\beta^{(\ell)} \right\|_2^2 + \sum_{j=1}^{d} \frac{1}{2\omega_j} \sum_{\ell=1}^{m} (\beta_j^{(\ell)})^2 + \frac{\mu}{2\sigma^2} \sum_{j=1}^{d} \omega_j \right],$$

with $\lambda = \sigma\sqrt{m}$, where $\beta = \{\beta^{(\ell)}\}_{\ell=1,\ldots,m}$ and $\omega = \{\omega_j\}_{j=1,\ldots,d}$, is a special case of

$$\min_{\beta,\Omega\in\mathcal{S}} \left[\sum_{\ell=1}^{m} \frac{1}{2\sigma^2} \left\| Y^{(\ell)} - X^{(\ell)}\beta^{(\ell)} \right\|_2^2 + \frac{1}{2} \sum_{\ell=1}^{m} (\beta^{(\ell)})^\top \Omega^{-1} \beta^{(\ell)} + \mu\Phi(\Omega) \right],$$

where $\mu > 0$ is a tuning parameter, Ω is a hyper parameter covariance matrix shared among the tasks, and $\mathcal{S}$ is a subset of symmetric positive semidefinite matrices. $\Phi(\Omega)$ is a penalty function for Ω with the goal of providing a good estimate of the covariance matrix Ω in the common quadratic regularizer $(\beta^{(\ell)})^\top \Omega^{-1} \beta^{(\ell)}$ shared by all the tasks. If $\Phi(\Omega)$ is a convex function of Ω, then (81) is jointly convex in $[\beta, \Omega]$ because it is well-known (and easy to verify) that $(\beta^{(\ell)})^\top \Omega^{-1} \beta^{(\ell)}$ is jointly convex in $[\beta^{(\ell)}, \Omega]$.

The group Lasso formulation (80) assumes a specific form of diagonal covariance matrix class $\mathcal{S} = \{\Omega = \text{diag}(\{\omega_j\}) : \omega_j \geq 0\}$. The specific choice of $\Phi(\Omega)$ is the simple L_1 penalty function $\Phi(\Omega) = \sum_j \omega_j$ that encourages sparsity. Unfortunately, this simple choice is suboptimal, as shown previously. It was also shown that it is possible to achieve better estimation of β by using a more sophisticated convex penalty function $\Phi(\Omega)$.The joint model presented in this section subsumes all the models described previously. It is worth mentioning that the Bayesian approach of [119] can

also be regarded as a special case of (81) but with a non-convex $\Phi(\Omega)$. However, due to the usual local minimum issues associated with nonconvex formulations, in practice this method does not perform as well as the convex formulations proposed in this chapter.

CHAPTER VI

SMOOTH SPARSE CODING

6.1 Introduction

Sparse coding is a popular unsupervised paradigm for learning sparse representations of data samples that are subsequently used in classification tasks. In standard sparse coding, each data sample is coded independently with respect to the dictionary. We propose a smooth alternative to traditional sparse coding that incorporates feature similarity, temporal or other user-specified domain information between the samples into the coding process.

The idea of smooth sparse coding is motivated by the relevance weighted likelihood principle. Our approach constructs a code that is efficient in a smooth sense and as a result leads to improved statistical accuracy over traditional sparse coding. The smoothing operation, which can be expressed as non-parametric kernel smoothing, provides a flexible framework for incorporating several types of domain information that might be available for the user. For example, in image classification task, one could use: (1) kernels in feature space for encoding similarity information for images and videos, (2) kernels in time space in case of videos for incorporating temporal relationship, and (3) kernels on unlabeled image in the semi-supervised learning and transfer learning settings.

Most sparse coding training algorithms fall under the general category of alternating procedures with a convex lasso regression sub-problem. While efficient algorithms for such cases exist [66], their scalability for large dictionaries remains a challenge. We propose a novel training method for sparse coding based on marginal regression, rather than solving the traditional alternating method with lasso sub-problem.

Marginal regression corresponds to several univariate linear regression followed by a thresholding step to promote sparsity. For large dictionary sizes, this leads to a dramatic speedup compared to traditional sparse coding methods (up to two orders of magnitude) without sacrificing statistical accuracy.

We also develop theory that extends the sample complexity result of [115] for dictionary learning using standard sparse coding to the smooth sparse coding case. This result specifically shows how the sample complexity depends on the L_1 norm of the kernel function used.

Our main contributions in this chapter are: (1) proposing a framework based on kernel-smoothing for incorporating feature, time or other similarity information between the samples into sparse coding, (2) providing sample complexity results for dictionary learning using smooth sparse coding, (3) proposing an efficient marginal regression training procedure for sparse coding, and (4) successful application of the proposed method in various classification tasks. Our contributions lead to improved classification accuracy in conjunction with computational speedup of two orders of magnitude.

6.2 Related work

Our approach is related to the local regression method [71, 52]. More recent related work is [77] that uses smoothing techniques in high-dimensional lasso regression in the context of temporal data. Another recent approach proposed by [125] achieves code locality by approximating data points using a linear combination of nearby basis points. The main difference is that traditional local regression techniques do not involve basis learning. In this work, we propose to learn the basis or dictionary along with the regression coefficients locally.

In contrast to previous sparse coding works we propose to use marginal regression for learning the regression coefficients, which results in a significant computational

speedup with no loss of accuracy. Marginal regression is a relatively old technique that has recently reemerged as a computationally faster alternative to lasso regression [39]. See also [45] for a statistical comparison of lasso regression and marginal regression.

6.3 Smooth Sparse Coding

In this chapter, the notation $|f|_p$ corresponds to the L_p norm of the function f: $(\int |f|^p \, d\mu)^{1/p}$. The standard sparse coding problem consists of solving the following optimization problem,

$$\begin{aligned} \min_{\substack{D \in \mathbb{R}^{d \times K} \\ \beta_i \in \mathbb{R}^K, i=1,\ldots,n}} & \sum_{i=1}^{n} \|X^{(i)} - D\beta_i\|_2^2 \\ \text{subject to} \quad & \|d_j\|_2 \leq 1 \quad j = 1, \ldots K \\ & \|\beta_i\|_1 \leq \lambda \quad i = 1, \ldots n. \end{aligned}$$

where $\beta_i \in \mathbb{R}^K$ corresponds to the encoding of sample $X^{(i)}$ with respected to the dictionary $D \in \mathbb{R}^{d \times K}$ and $d_j \in \mathbb{R}^d$ denotes the j-column of the dictionary matrix D. The dictionary is typically over-complete, implying that $K > d$.

Object recognition is a common sparse coding application where $X^{(i)}$ corresponds to a set of features obtained from a collection of image patches, for example SIFT features [73]. The dictionary D corresponds to an alternative coding scheme that is higher dimensional than the original feature representation. The L_1 constraint promotes sparsity of the new encoding with respect to D. Thus, every sample is now encoded as a sparse vector that is of higher dimensionality than the original representation.

In some cases the data exhibits a structure that is not captured by the above sparse coding setting. For example, SIFT features corresponding to samples from the same class are presumably closer to each other compared to SIFT features from other classes. Similarly in video, neighboring frames are presumably more related to each other than frames that are farther apart. In this chapter we propose a mechanism

to incorporate such feature similarity and temporal information into sparse coding, leading to a sparse representation with an improved statistical accuracy (for example as measured by classification accuracy).

We consider the following smooth version of the sparse coding problem above:

$$\min_{\substack{D\in\mathbb{R}^{d\times K}\\ \beta_i\in\mathbb{R}^K, i=1,\ldots,n}} \sum_{i=1}^{n}\sum_{j=1}^{n} w(X^{(j)}, X^{(i)})\|X^{(j)} - D\beta_i\|_2^2 \tag{91}$$

$$\text{subject to} \qquad \|d_j\|_2 \leq 1 \quad j = 1,\ldots K \tag{92}$$

$$\|\beta_i\|_1 \leq \lambda \quad i = 1,\ldots n. \tag{93}$$

where $\sum_{j=1}^{n} w(X^{(j)}, X^{(i)}) = 1$ for all i. It is convenient to define the weight function through a smoothing kernel

$$w(X^{(j)}, X^{(i)}) = \frac{1}{h_1}\mathcal{K}_1\left(\frac{\rho(X^{(j)}, X^{(i)})}{h_1}\right)$$

where $\rho(\cdot,\cdot)$ is a distance function that captures the feature similarity, h_1 is the bandwidth, and $\mathcal{K}_1$ is a smoothing kernel. Traditional sparse coding minimizes the reconstruction error of the encoded samples. Smooth sparse coding, on the other hand, minimizes the reconstruction of encoded samples with respect to their neighbors (weighted by the amount of similarity).

The smooth sparse coding setting leads to codes that represent a neighborhood rather than an individual sample and that have lower mean square reconstruction error (with respect to a given dictionary), due to lower estimation variance (see for example the standard theory of smoothed empirical process [32]). There are several possible ways to determine the weight function w. One common choice for the kernel function is the Gaussian kernel whose bandwidth is selected using cross-validation. Other common choices for the kernel are the triangular, uniform, and tricube kernels. The bandwidth may be fixed throughout the input space, or may vary in order to take advantage of non-uniform samples. We use in our experiment the tricube kernel with a constant bandwidth.

The distance function $\rho(\cdot, \cdot)$ may be one of the standard distance functions (for example based on the L_p norm). Alternatively, $\rho(\cdot, \cdot)$ may be expressed by domain experts, learned from data before the sparse coding training, or learned jointly with the dictionary and codes during the sparse coding training.

6.3.1 Spatio-Temporal smoothing

In spatio-temporal applications we can extend the kernel to include also a term reflecting the distance between the corresponding time or space

$$w(X^{(j)}, X^{(i)}) = \frac{1}{h_1}\mathcal{K}_1\left(\frac{\rho(X^{(j)}, X^{(i)})}{h_1}\right)\frac{1}{h_2}\mathcal{K}_2\left(\frac{j-i}{h_2}\right).$$

Above, $\mathcal{K}_2$ is a univariate symmetric kernel with bandwidth parameter h_2. One example is video sequences, where the kernel above combines similarity of the frame features and the time-stamp.

Alternatively, the weight function can feature only the temporal component and omit the first term containing the distance function between the feature representation. A related approach for that situation, is based on the Fused lasso which penalizes the absolute difference between codes for neighboring points. The main drawback of that approach is that one needs to fit all the data points simultaneously whereas in smooth sparse coding, the coefficient learning step decomposes as n separate problems which provides a computational advantage (see supplementary document for more details). Also, while fused Lasso penalty is suitable for time-series data to capture relatedness between neighboring frames, it may not be immediately suitable for other situations that the proposed smooth sparse coding method could handle.

6.4 Marginal Regression for Smooth Sparse Coding

A standard algorithm for sparse coding is the alternating bi-convex minimization procedure, where one alternates between (i) optimizing for codes (with a fixed dictionary)

and (ii) optimizing for dictionary (with fixed codes). Note that step (i) corresponds to regression with L_1 constraints and step (ii) corresponds to least squares with L_2 constraints. In this section we show how marginal regression could be used to obtain better codes faster (step (i)). In order to do so, we first give a brief description of the marginal regression procedure.

Marginal Regression: Consider a regression model $Y = X\beta + z$ where $y \in \mathbb{R}^n$, $\beta \in \mathbb{R}^p$, $X \in \mathbb{R}^{n \times p}$ with L_2 normalized columns (denoted by $X^{(j)}$), and z is the noise vector. Marginal regression proceeds as follows:

- Calculate the least squares solution

$$\hat{\alpha}^{(j)} = {X^{(j)}}^T y.$$

- Threshold the least-square coefficients

$$\hat{\beta}^{(j)} = \hat{\alpha}^{(j)} 1_{\{|\hat{\alpha}^{(j)}| > t\}}, \quad j = 1, \ldots, p.$$

Marginal regression requires just $O(np)$ operations compared to $O(p^3 + np^2)$, the typical complexity of lasso algorithms. When p is much larger than n, marginal regression provides two orders of magnitude speedup over Lasso based formulations. Note that in sparse coding, the above speedup occurs for each iteration of the outer loop, thus enabling sparse coding for significantly larger dictionary sizes. Recent studies have suggested that marginal regression is a viable alternative for Lasso given its computational advantage over lasso. A comparison of the statistical properties of marginal regression and lasso is available in [40, 45].

Code update (step (i)): Applying marginal regression to smooth sparse coding, we obtain the following scheme. The marginal least squares coefficients are

$$\hat{\alpha}_i^{(k)} = \sum_{j=1}^{n} \frac{w(X^{(j)}, X^{(i)})}{\|d_k\|_2} d_k^T X^{(j)}.$$

We sort these coefficient in terms of their absolute values, and select the top s coefficients whose L_1 norm is bounded by λ:

$$\hat{\beta}_i^{(k)} = \begin{cases} \hat{\alpha}_i^{(k)} & k \in S \\ 0 & k \notin S \end{cases}, \quad \text{where}$$

$$S = \left\{ 1, \ldots, s \,:\, s \leq d : \sum_{k=1}^{s} |\hat{\alpha}_i^{(k)}| \leq \lambda \right\}$$

We select the thresholding parameter using cross validation in each of the sparse coding iterations. Note that the same approach could be used with structured regularizers too, for example [21, 60].

Dictionary update (step (ii)): Marginal regression works well when there is minimal correlation between the different dictionary atoms. In the linear regression setting, marginal regression performs much better with orthogonal data [45]. In the context of sparse coding, this corresponds to having uncorrelated or incoherent dictionaries [111]. One way to measure such incoherence is using the babel function, which bounds the maximum inner product between two different columns d_i, d_j:

$$\mu_s(D) = \max_{i \in \{1,\ldots,d\}} \max_{\Lambda \subset \{1,\ldots,d\} \setminus \{i\}; |\Lambda| = s} \sum_{j \in \Lambda} |d_j^\top d_i|.$$

An alternative, which leads to easier computation is by adding the term $\|D^T D - I_{K \times K}\|_F^2$ to the reconstruction objective, when optimizing over the dictionary matrix D. This leads to the following optimization problem for dictionary update step:

$$\hat{D} = \arg\min_{D \in \mathcal{D}} F(D) \quad \text{where}$$

$$F(D) = \sum_{i=1}^{n} \|X^{(i)} - D\hat{\beta}_i\|_2^2 + \gamma \|D^\top D - I\|_F^2$$

and $\mathcal{D} = \{D \in \mathbb{R}^{d \times K} : \|d_j\|_2 \leq 1\}$. The regularization term γ controls the level of incoherence enforced.

This optimization problem is of the form of minimizing a differentiable function over a closed convex set. We use the gradient projection method [13, 98] for solving

the above optimization problem. The gradient (cf. [75]) of the above expression with respect to D at each iteration is given by $\nabla F(D) = 2\left(D\hat{B}\hat{B}^\top - X\hat{B}^\top\right) + 4\gamma\left(DD^\top D - D\right)$, where $\hat{B} = [\hat{\beta}_1, \ldots, \hat{\beta}_n]$ is the matrix of codes from the previous code update step, $X \in \mathbb{R}^{p\times n}$ is the data in matrix format. The gradient projection descent iterations are given by

$$D(t+1) = \Pi_{\mathcal{D}}\left(D(t) - \eta_t \nabla F(D(t))\right).$$

where by $\Pi_{\mathcal{D}}$, we denote column-wise projection of the dictionary matrix on to the unit ball and t is the index for sub-iteration count for each dictionary update step. Specifically, for each dictionary update step, we run the gradient projected descent algorithm untill convergence (more details about this in experimental section). Note that projection of a vector onto the l_2 ball is straightforward since we only need to rescale the vector towards the origin, i.e., normalize the vectors with length greater than 1.

Convergence to local point of gradient projection methods for minimizing differentiable functions over convex set have been analyzed in [98]. Similar guarantees could be provided for each of the dictionary update steps. A heuristic approach for dictionary update with incoherence constraint was proposed in [83] and more recently in [93](where the L-BFGS method was used for the unconstrained problem and the norm constraint was enforced at the final step). We found that the proposed gradient projected descent method performed empirically better than both the approaches. Furthermore both approaches are heuristic and do not guarantee local convergence for the dictionary update step.

Finally, a sequence of such updates corresponding to step (i) and step (ii) converges to a stationary point of the optimization problem (this can be shown using Zangwill's theorem [127]). But no provable algorithm that converges (under certain assumptions) to the global minimum of the smooth sparse coding (or standard sparse

Algorithm 4 Smooth Sparse Coding via Marginal Regression

Input: Data $\{(X^{(1)}, Y^{(1)}), \ldots, (X^{(n)}, Y^{(1)})\}$ and kernel/similarity measure $\mathcal{K}_1$ and d_1.

Precompute: Compute the weight matrix $w(i,j)$ using the kernel/similarity measure

Initialize: Set the dictionary at time zero to be D_0.

Algorithm:

repeat

Step (i): For all $i = 1, \ldots, n$, solve marginal regression:

$$\hat{\alpha}_i^{(k)} = \sum_{j=1}^{n} \frac{w(X^{(j)}, X^{(i)})}{\|d_k\|_2} d_k^T X^{(j)}$$

$$\hat{\beta}_j^{(k)} = \begin{cases} \hat{\alpha}_j^{(k)} & j \in S \\ 0 & j \notin S \end{cases},$$

$$S = \{1, \ldots, s; s \leq d : \sum_{k=1}^{s} |\hat{\alpha}_i^{(k)}| \leq \lambda\}.$$

Step (ii): Update the dictionary based on codes from previous step.

$$\hat{D} = \arg\min_{D \in \mathcal{D}} \sum_{i=1}^{n} \|X^{(i)} - D\hat{\beta}_i\|_2^2 + \gamma \|D^\top D - I\|_F^2$$

$$\text{where } \mathcal{D} = \{D \in \mathbb{R}^{d \times K} : \|d_j\|_2 \leq 1\}$$

until convergence

Output: Return the learned codes and dictionary.

coding) exists yet. Nevertheless, the main idea of this section is to speed-up the existing alternating minimization procedure for obtaining sparse representations, by using marginal regression. We leave a detailed theoretical analysis of the individual dictionary update steps and the overall alternating procedure (for codes and dictionary) as future work.

6.5 Sample Complexity of Smooth sparse coding

In this section, we analyze the sample complexity of the proposed smooth sparse coding framework. Specifically, since there does not exist a provable algorithm that converges to the global minimum of the optimization problem in Equation (91), we

provide uniform convergence bounds over the dictionary space and thereby prove a sample complexity result for dictionary learning under smooth spare coding setting. We leverage the analysis for dictionary learning in the standard sparse coding setting by [115] and extend it to the smooth sparse coding setting. The main difficulty for the smooth sparse coding setting is obtaining a covering number bound for an appropriately defined class of functions (see Theorem 1 for more details).

We begin by re-representing the smooth sparse coding problem in a convenient form for analysis. Let $x_1, \ldots, x_n$ be independent random variables with a common probability measure $\mathbb{P}$ with a density p. We denote by $\mathbb{P}_n$ the empirical measure over the n samples, and the kernel density estimate of p is defined by

$$p_{n,h}(x) = \frac{1}{nh} \sum_{i=1}^{n} \mathcal{K}\left(\frac{\|x - X^{(i)}\|_2}{h}\right).$$

Let $\mathcal{K}_{h_1}(\cdot) = \frac{1}{h_1}\mathcal{K}_1(\frac{\cdot}{h})$. With the above notations, the reconstruction error at the point x is given by

$$r_\lambda(x) = \int \min_{\beta \in \mathcal{S}_\lambda} \|x' - D\beta\|_2 \mathcal{K}_{h_1}(\rho(x, x'))\, d\mathbb{P}_n(x')$$

where

$$\mathcal{S}_\lambda = \{\beta : \|\beta\|_1 \leq \lambda\}.$$

The empirical reconstruction error is

$$\mathsf{E}_{\mathbb{P}_n}(r) = \iint \min_{\beta \in \mathcal{S}_\lambda} \|x' - D\beta\|_2 \mathcal{K}_{h_1}(\rho(x, x'))\, d\mathbb{P}_n(x')\, dx$$

and its population version is

$$\mathsf{E}_{\mathbb{P}}(r) = \iint \min_{\beta \in \mathcal{S}_\lambda} \|x' - D\beta\|_2 \mathcal{K}_{h_1}(\rho(x, x'))\, d\mathbb{P}(x')\, dx.$$

Our goal is to show that the sample reconstruction error is close to the true reconstruction error. Specifically, to show $\mathsf{E}_{\mathbb{P}}(r_\lambda) \leq (1 + \kappa)\mathsf{E}_{\mathbb{P}_n}(r_\lambda) + \epsilon$ where $\epsilon, \kappa \geq 0$, we bound the covering number of the class of functions corresponding to the reconstruction error. We assume a dictionary of bounded babel function, which holds as a

result of the relaxed orthogonality constraint used in the Algorithm 4 (see also [83]). We define the set of r functions with respect to the the dictionary D (assuming data lies in the unit d-dimensional ball $\mathbb{S}^{d-1}$) by

$$\mathcal{F}_\lambda = \{r_\lambda : \mathbb{S}^{d-1} \to \mathbb{R} : D \in \mathbb{R}^{d\times K}, \|d_i\|_2 \leq 1, \mu_s(D) \leq \gamma\}.$$

The following theorem bounds the covering number of the above function class.

Theorem 6.5.1. *For every $\epsilon > 0$, the metric space $(\mathcal{F}_\lambda, |\cdot|_\infty)$ has a subset of cardinality at most $\left(\frac{4\lambda|\mathcal{K}_{h_1}(\cdot)|_1}{\epsilon(1-\gamma)}\right)^{dK}$, such that every element from the class is at a distance of at most ϵ from the subset, where $|\mathcal{K}_{h_1}(\cdot)|_1 = \int |\mathcal{K}_{h_1}(x)|\, d\mathbb{P}$.*

Proof. Let $\mathcal{F}'_\lambda = \{r'_\lambda : \mathbb{S}^{d-1} \to \mathbb{R} : D \in d \times K, \|d_i\|_2 \leq 1\}$, where $r'_\lambda(x) = \min_{\beta\in\mathcal{S}_\lambda} \|D\beta - x\|$. With this definition we note that $\mathcal{F}_\lambda$ is just $\mathcal{F}'_\lambda$ convolved with the kernel $\mathcal{K}_{h_1}(\cdot)$. By Young's inequality [32] we have,

$$|\mathcal{K}_{h_1} * (s_1 - s_2)|_p \leq |\mathcal{K}_{h_1}|_1 |s_1 - s_2|_p, \quad 1 \leq p \leq \infty$$

for any L_p integrable functions s_1 and s_2. Using this fact, we see that convolution mapping between metric spaces $\mathcal{F}'$ and $\mathcal{F}$ converts $\frac{\epsilon}{|\mathcal{K}_{h_1}(\cdot)|_1}$ covers into ϵ covers. From [115], we have that the class $\mathcal{F}'_\lambda$ has ϵ covers of size at most $(\frac{4\lambda}{\epsilon(1-\gamma)})^{dK}$. This proves the statement of the theorem. □

The above theorem could be used in conjunction with standard statements in the literature for bounding the generalization error of empirical risk minimization algorithms based on covering numbers. We have provided the general statements in the appendix for completeness of this chapter. The proofs of the general statements could be found in the references cited. Below, we provide two such generalization bounds for smooth sparse coding problem, corresponding to slow rates and fast rates.

Slow rates: When the theorem on covering numbers for the function class $\mathcal{F}_\lambda$ (Theorem 6.5.1) is used along with Lemma 1 stated in the appendix (corresponding to slow rate generalization bounds) it is straightforward to obtain the following generalization bounds with slow rates for the smooth sparse coding problem.

Theorem 6.5.2. *Let $\gamma < 1$, $\lambda > e/4$ with distribution $\mathbb{P}$ on $\mathbb{S}^{d-1}$. Then with probability at least $1 - e^{-t}$ over the n samples drawn according to $\mathbb{P}$, for all the D with unit length columns and $\mu_s(D) \leq \gamma$, we have:*

$$\mathbf{E}_{\mathbb{P}}(r_\lambda) \leq \mathbf{E}_{\mathbb{P}_n}(r_\lambda) + \sqrt{\frac{dK \ln\left(\frac{4\sqrt{n}\lambda|\mathcal{K}_{h_1}(\cdot)|_1}{(1-\gamma)}\right)}{2n}} + \sqrt{\frac{t}{2n}} + \sqrt{\frac{4}{n}}$$

The above theorem, establishes that the generalization error scales as $O(n^{-1/2})$ (assuming the other problem parameters are fixed).

Fast rates: Under further assumptions ($\kappa > 0$), it is possible to obtain faster rates of $O(n^{-1})$ for smooth sparse coding, similar to the ones obtained for general learning problems in [8]. The following theorem gives the precise statement.

Theorem 6.5.3. *Let $\gamma < 1$, $\lambda > e/4$, $dK > 20$ and $n \geq 5000$. Then with probability at least $1 - e^{-t}$, we have for all D with unit length and $\mu_s(D) \leq \gamma$,*

$$\mathbf{E}_{\mathbb{P}}(r_\lambda) \leq 1.1\mathbf{E}_{\mathbb{P}_n}(r_\lambda) + 9\frac{dK \ln\left(\frac{4n\lambda|\mathcal{K}_{h_1}(\cdot)|_1}{(1-\gamma)}\right) + t}{n}.$$

The above theorem follows from the theorem on covering number bound (Theorem 6.5.1) above and Lemma 2 from the appendix. In both statements the definition of $r_\lambda(x)$ differs from (1) by a square term, but it could easily be incorporated into the above bounds resulting in an additive factor of 2 inside the logarithm term as is done in [115].

6.6 Experiments

We demonstrate the proposed approach both in terms of speed-up and accuracy, over standard sparse coding. A detailed description of all real-world data sets used in the experiments are given in the appendix. As discussed before, the overall optimization procedure is non-convex. The stopping criterion was chosen as when the value of the reconstruction error did not change by more than 0.001%. Though this does not guarantee convergence to a global optimum, according to the experimental results,

we see that the points of convegence invariably resulted in a good local optimum (as reflected by the good empirical performance). Furthermore, in all the experiments, we ran 10 iterations of the projected gradient descent algorithm for each dictionary update step. We fixed the learning rate for all iterations of gradient projection descent algorithm as $\eta = \eta_t = 0.01$ as it was found to performed well in the experiments. The parameters γ and t are set for each experiment based on cross-validation (we first tuned for γ and then for t) for classification results on training set as is done in the literature [123].

6.6.1 Speed comparison

We conducted synthetic experiments to examine the speed-up provided by sparse coding with marginal regression. The data was generated from a 100 dimensional mixture of two Gaussian distribution that satisfies $\|\mu_1 - \mu_2\|_2 = 3$ (with identity covariance matrices). The dictionary size was fixed at 1024.

We compare the proposed smooth sparse coding algorithm, standard sparse coding with lasso [66] and marginal regression updates respectively, with a relative reconstruction error $\|X - \hat{D}\hat{B}\|_F / \|X\|_F$ convergence criterion. We experimented with different values of the relative reconstruction error (less than 10%) and report the average time. From Table 10, we see that smooth sparse coding with marginal regression takes significantly less time to achieve a fixed reconstruction error. This is due to the fact that it takes advantage of the spatial structure and use marginal regression updates. It is worth mentioning that standard sparse coding with marginal regression updates performs faster compared to the other two methods that uses lasso updates, as expected (but does not take into account the spatial structure).

6.6.2 Experiments with Kernel in Feature space

We conducted several experiments demonstrating the advantage of the proposed coding scheme in different settings. Concentrating on face and object recognition from

Table 10: Time comparison of coefficient learning in SC and SSC with either Lasso or Marginal regression updates. The dictionary update step was same for all methods.

Method	time (sec)
SC+LASSO	524.5 ±12
SC+MR	242.2±10
SSC+LASSO	560.2±12
SSC+MR	184.4 ±19

static images, we evaluated the performance of the proposed approach along with standard sparse coding and LLC [125], another method for obtaining sparse features based on locality. Also, we performed experiments on activity recognition from videos based on both space and time based kernels. As mentioned before all results are reported using tricube kernel.

6.6.2.1 Image classification

We conducted image classification experiments on CMU-multipie, 15 Scene and Caltech-101 data sets. Following [123] , we used the following approach for generating sparse image representation: we densely sampled 16×16 patches from images at the pixel level on a grid with step size 8 pixels, computed SIFT features, and then computed the corresponding sparse codes over a 1024-size dictionary. We used max pooling to get the final representation of the image based on the codes for the patches. The process was repeated with different randomly selected training and testing images and we report the average per-class recognition rates (together with its standard deviation estimate) based on one-vs-all SVM classification. As Table 11 indicates, our smooth sparse coding algorithm resulted in significantly higher classification accuracy than standard sparse coding and LLC. In fact, the reported performance is better than previous reported results using unsupervised sparse coding techniques [123].

Dictionary size: In order to demonstrate the use of scalability of the proposed method with respect to dictionary size, we report classification accuracy with increasing dictionary sizes using smooth sparse coding. The main advantage of the

Table 11: Test set error accuracy for face recognition on CMU-multipie data set (left) 15 scene (middle) and Caltech-101 (right) respectively. The performance of the smooth sparse coding approach is better than the standard sparse coding and LLC in all cases.

	CMU-multipie	15 scene	Caltech-101
SC	92.70±1.21	80.28±2.12	73.20±1.14
LLC	93.70±2.22	82.28±1.98	74.82±1.65
SSC	95.05 ±2.33	84.53±2.57	77.54±2.59

Table 12: Effect of dictionary size on classification accuracy using smooth sparse coding and marginal regression on 15 scene and Caltech -101 data set.

Dictionary size	15 scene	Caltech-101
1024	84.42±2.01	77.14 ±2.23
2048	87.92±2.35	79.75±1.44
4096	90.22±2.91	81.01±1.17

proposed marginal regression training method is that one could easily run experiments with larger dictionary sizes, which typically takes a significantly longer time for other algorithms. For both the Caltech-101 and 15-scene data set, classification accuracy increases significantly with increasing dictionary sizes as seen in Table 12.

6.6.2.2 Action recognition:

We further conducted an experiment on activity recognition from videos with KTH action and YouTube data set (see Appendix). Similar to the static image case, we follow the standard approach for generating sparse representations for videos as in [118]. We densely sample $16 \times 16 \times 10$ blocks from the video and extract HoG-3d [64] features from the sampled blocks. We then use smooth sparse coding and max-pooling to generate the video representation (dictionary size was fixed at 1024 and cross-validation was used to select the regularization and bandwidth parameters). Previous approaches include sparse coding, vector quantization, and k-means on top of the HoG-3d feature set (see [118] for a comprehensive evaluation). As indicated by Table 13, smooth sparse coding results in higher classification accuracy than previously reported state-of-the-art and standard sparse coding on both datasets (see

Table 13: Action recognition (accuracy) for cited method (left), Hog3d+ SC (middle) and Hog3d+ SSC (right): KTH data set(top) YouTube action dataset (bottom).

Cited method	SC	SSC
92.10 [118]	92.423	94.393
71.2 [70]	72.640	75.022

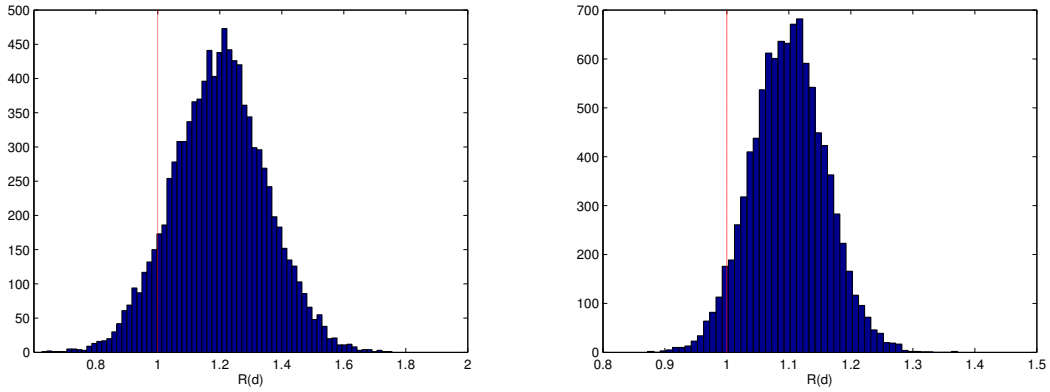


Figure 27: Comparison between the histograms of Fisher discriminant score realized by sparse coding and smooth sparse coding. The images represent the histogram of the ratio of smooth sparse coding Fisher score over standard sparse coding Fisher score (left: image data set; right: video). A value greater than 1 implies that smooth sparse coding is more discriminatory.

[118, 70] for a description of the alternative techniques).

6.6.2.3 Discriminatory power

In this section, we describe another experiment that contrasts the codes obtained by sparse coding and smooth sparse coding in the context of a subsequent classification task. As in [124], we first compute the codes in both case based on patches and combine it with max-pooling to obtain the image level representation. We then compute the fisher discriminant score (ratio of within-class variance to between-class variance) for each dimension as measures of the discrimination power realized by the representations.

Figure 27, graphs a histogram of the ratio of smooth sparse coding Fisher score over standard sparse coding Fisher score $R(d) = F_1(d)/F_2(d)$ for 15-scene dataset

(left) and Youtube dataset (right). Both histograms demonstrate the improved discriminatory power of smooth sparse coding over regular sparse coding.

6.7 Semi-supervised smooth sparse coding

One of the primary difficulties in some image classification tasks is the lack of availability of labeled data and in some cases, both labeled and unlabeled data (for particular domains). This motivated semi-supervised learning and transfer learning without labels [82] respectively. The motivation for such approaches is that data from a related domain might have some visual patterns that might be similar to the problem at hand. Hence, learning a high-level dictionary based on data from a different domains aids the classification task of interest.

We propose that the smooth sparse coding approach might be useful in this setting. The motivation is as follows: in semi-supervised, typically not all samples from a different data set might be useful for the task at hand. Using smooth sparse coding, one can weigh the useful points more than the other points (the weights being calculated based on feature/time similarity kernel) to obtain better dictionaries and sparse representations. Other approach to handle a lower number of labeled samples include collaborative modeling or multi-task approaches which impose a shared structure on the codes for several tasks and use data from all the tasks simultaneously, for example group sparse coding [10]. The proposed approach provides an alternative when such collaborative modeling assumptions do not hold, by using relevant unlabeled data samples that might help the task at hand via appropriate weighting.

We now describe an experiment that examines the proposed smoothed sparse coding approach in the context of semi-supervised dictionary learning. We use data from both CMU multi-pie dataset (session 1) and faces-on-tv dataset (treated as frames) to learn a dictionary using a feature similarity kernel. We follow the same procedure described in the previous experiments to construct the dictionary. In the

test stage we use the obtained dictionary for coding data from sessions 2, 3, 4 of CMU-multipie data set, using smooth sparse coding. Note that semi-supervision was used only in the dictionary learning stage (the classification stage used supervised SVM).

Table 14 shows the test set error rate and compares it to standard sparse coding and LLC [125]. Smooth sparse coding achieves significantly lower test error rate than the two alternative techniques. We conclude that the smoothing approach described in this chapter may be useful in cases where there is a small set of labeled data, such as semisupervised learning and transfer learning.

Table 14: Semi-supervised learning test set error: Dictionary learned from both CMU multi-pie and faces-on-tv data set using feature similarity kernel, used to construct sparse codes for CMU multipie data set.

Method	SC	LLC	SSC-tricube
Test errror	6.345	6.003	4.975

6.8 Data set Description

6.8.1 CMU Multi-pie face recognition:

The face recognition experiment was conducted on the CMU Multi-PIE dataset. The dataset is challenging due to the large number of subjects and is one of the standard data sets used for face recognition experiments. The data set contains 337 subjects across simultaneous variations in pose, expression, and illumination. We ignore the 88 subjects that were considered as outliers in [123] and used the rest of the images for our face recognition experiments. We follow [123] and use the 7 frontal extreme illuminations from session one as train set and use other 20 illuminations from Sessions 2-4 as test set.

6.8.2 15 Scenes Categorization:

We also conducted scene classification experiments on the 15-Scenes data set. This data set consist of 4485 images from 15 categories, with the number of images each

category ranging from 200 to 400. The categories corresponds to scenes from various settings like kitchen, living room etc. Similar to the previous experiment, we extracted patches from the images and computed the SIFT features corresponding to the patches. The categorization results are reported in Table 2. The accuracy using smooth sparse codes is better than previous reported results on this data set using standard sparse coding techniques for e.g., [122].

6.8.3 Caltech-101 Data set:

The Caltech-101 data set consists of images from 101 classes like animals, vehicles, flowers, etc. The number of images per category varies from 30 to 800. Most images are of medium resolution (300×300). All images are used a gray-scale images. Following previous standard experimental settings for Caltech-101 data set, we use 30 images per category and test on the rest. Average classification accuracy normalized by class frequency is used for evaluation. Similar to the previous experiment, we extracted patches from the images and computed the SIFT features corresponding to the the patches. Table 2 shows the accuracy of sparse coding and smooth sparse coding. Note that sparse coding on SIFT achieves one of the best results on the Caltech-101 data set. The proposed smoothing approach further improves the accuracy and achieves competitive results on this benchmark data set.

6.8.4 Activity recognition

The KTH action dataset consists of 6 human action classes. Each action is performed several times by 25 subjects and is recorded in four different scenarios. In total, the data consists of 2391 video samples. The YouTube actions data set has 11 action categories and is more complex and challenging [70]. It has 1168 video sequences of varied illumination, background, resolution etc. We randomly densely sample blocks (400 cuboids) of video from the data sample and extract HOG-3d features and constructed the video features as described above. .

6.8.5 Youtube person data set

Similar to the experiments using the feature smoothing kernel, in this section we report results on experiment conducted using the time smoothed kernel. Specifically, we used the YouTube person data set [63] in order to recognize people, based on time-based kernel smooth sparse coding. The dataset contains 1910 sequences of 47 subjects. The approach for this experiment is similar to [122]. We extracted SIFT descriptors for every 16×16 patches sampled on a grid of step size 8. Then we use smooth sparse coding with time kernel to learn the codes and max pooling to get the final representation of a video sample. Pre-processing steps like face extraction or face tracking was not used in this experiment. Finally, linear svm was used for classification of video sequences based on person present in the video sequences.

6.9 Experiments using Temporal Smoothing

In this section we describe an experiment conducted using the temporal smoothing kernel on the Youtube persons dataset. We extracted SIFT descriptors for every 16×16 patches sampled on a grid of step size 8 and used smooth sparse coding with time kernel to learn the codes and max pooling to get the final video representation. We avoided pre-processing steps such as face extraction or face tracking. Note that in the previous action recognition video experiment, video blocks were densely sampled and used for extracting HoG-3d features. In this experiment, on the other hand, we extracted SIFT features from individual frames and used the time kernels to incorporate the temporal information into the sparse coding process.

Table 15: Linear SVM accuracy for person recognition task from YouTube face video dataset.

Method	Fused Lasso	SC	SSC-tricube
Accuracy	68.59	65.53	71.21

For this case, we also compared to the more standard fused-lasso based approach [109]. Note that in fused Lasso based approach, in addition to the standard L_1 penalty, an additional L_1 penalty on the difference between the neighboring frames for each dimensions is used. This tries to enforce the assumption that in a video sequence, neighboring frames are more related to one another as compared to frames that are farther apart.

Table 15 shows that smooth sparse coding achieved higher accuracy than fused lasso and standard sparse coding. Smooth sparse coding has comparable accuracy on person recognition tasks to other methods that use face-tracking, for example [63]. Another advantage of smooth sparse coding is that it is significantly faster than sparse coding and the fused lasso.

6.10 Generalization bounds for learning problems

In this section, for completeness, we provide two generalization bounds for learning problems, corresponding to slow-rates and fast rates, based on covering numbers. We first state the following general lemma regarding generalization error bounds with slow rates for a learning problem with given covering number bounds.

Lemma 3 ([115])**.** *Let $\mathcal{Q}$ be a function class of $[0, B]$ functions with covering number $(\frac{C}{\epsilon})^d > \frac{e}{B^2}$ under $|\cdot|_\infty$ norm. Then for every $t > 0$ with probability at least $1 - e^{-t}$, for all $f \in \mathcal{Q}$, we have:*

$$\mathsf{E} f \leq E_n f + B\left(\sqrt{\frac{d\ln(C\sqrt{n})}{2n}} + \sqrt{\frac{t}{2n}}\right) + \sqrt{\frac{4}{n}}.$$

Next, we state a general lemma regarding generalization error bounds with fast rates

Lemma 4 ([115])**.** *Let $\mathcal{Q}$ be a function class of $[0, 1]$ functions that can be covered for any $\epsilon > 0$ by at most $(C/\epsilon)^d$ balls of radius ϵ in the $|\cdot|_\infty$ metric, where $C \geq e$ and*

$\beta > 0$. Then with probability at least $1 - \exp(-t)$ we have for all functions $f \in \mathcal{Q}$,

$$\mathsf{E} f \leq (1+\beta) E_n f + K(d, m, \beta) \frac{d \ln(Cm) + t}{n},$$

where $K(d, m, \beta) = \sqrt{2\left(\frac{9}{\sqrt{n}} + 2\right)\left(\frac{d+3}{3d}\right) + 1} + \left(\frac{9}{\sqrt{n}} + 2\right) + \left(\frac{d+3}{3d}\right) + 1 + \frac{1}{2\beta}$.

Note that $K(d, m, \beta)$ is non-increasing in d, m as a consequence of which we immediately have the following corollary, which we use in the statement of our main theorem for fast rates.

Corollary 6. *Let $\mathcal{Q}$ be as above. For $d \geq 20$, $m \geq 5000$ and $\beta = 0.1$, we have with probability at least $1 - \exp(-t)$ for all functions $f \in \mathcal{Q}$,*

$$Ef \leq (1.1) E_n f + 9 \frac{d \ln(Cm) + t}{n}.$$

The proofs of Lemma 1 and Lemma 2 could be found in [115]. Obtaining generalization bounds for the problem under consideration follows directly, given the above two general statements and our theorem on covering numbers (Theorem 1).

CHAPTER VII

FEATURE SCREENING VIA RKHS EMBEDDINGS

7.1 Introduction

Ultrahigh dimensional data sets are ubiquitous in modern statistical problems arising from several diverse scientific fields. For example, several biological problems or high frequency trading problems have several million features (denoted as d) compared to a much lesser number of samples (denoted as n). Feature screening plays an important role in analyzing these 'large d small n' data sets. Various penalization based techniques that promote sparsity have been developed and analyzed in this regime: Lasso [108], Dantzig selector [22] and scad penalties [38] assume a linear model between the covariates and the response, while SPAM and related techniques [84, 56] assume a non-linear model in order to select a few relevant features. All these methods allow for the data dimensionality to be greater than the sample size.

However, there are several issues with the above mentioned penalty approaches in ultrahigh dimensions. First, these methods cannot efficiently handle ultrahigh dimensional settings with d growing faster than a polynomial rate in n, e.g., d growing exponential in n. Second, the irrepresentability conditions [130]—these conditions mean that the covariates not in the true model are not representable, in some sense, by the covariates in the true model—under which the model selection consistency is proved for the penalty methods in high-dimensions, are too stringent to hold in ultrahigh dimensions [41](Section 5.5 for general discussion and concrete examples). Third, penalization approaches are computationally expensive, e.g., typical lasso algorithms scales as $O(d^3)$ and are therefore expensive for ultrahigh dimensional problems.

In order to tackle this situation, an alternate line of research based on marginal

regression was proposed and analyzed [39, 42]. This is a relatively old technique, that has re-emerged as an alternative for feature screening in ultrahigh dimensions. The general idea of this approach is to measure the relationship (to be clearly defined based on context) of each feature individually to the response and rank them accordingly. For example, assuming a linear model between response and covariates, [39] proposed to measure the residual between response and each covariate (in a least-square sense) and rank the covariates accordingly. In order to relax the linear model assumption, [42] proposed screening for generalized linear models based on marginal utility; [37] proposed screening using a non-parametric additive model based on smoothing splines. Recently, [68] proposed a model-free (i.e., without any regressive modeling assumptions) screening procedure, DC-SIS, based on distance covariance metric [104]—which is zero if and only if the random variables are independent—as a measure of relationship between response and covariate. To elaborate, if the distance covariance between the response and a covariate is "small", then the response is independent of the covariate and therefore such a covariate can be screened out from consideration. Recently [61] showed that a two-step procedure, screening followed by penalized regression, is optimal for feature selection in this regime.

In this chapter, we propose a general framework, *sup*-HSIC-SIS (Hilbert Schmidt independence criterion–Sure independence screening), for model-free, multi-output screening. The approach uses RKHS based independence measures [49] and generalizes the previously proposed DC-SIS approach. This proposal is motivated from the recent work by [90] that established an equivalence between distance covariance and HSIC (a dependence/independence measure based on RKHS embedding of probabilities). Given this equivalence, it is straightforward to propose an independence screening procedure based on HSIC by carrying out the analysis with HSIC replacing the distance covariance in [68]. However, a major issue with DC-SIS (or its equivalent RKHS version, say HSIC-SIS) is that the employed independence measure is

just one member of a parametric family of independence measures and there is no guarantee that this member provides the best screening procedure over all the other choices from this family. For example, if we consider HSIC-SIS, the choice of kernel determines the performance of the screening procedure.

Our main contribution in this chapter is to address this issue by using an independence measure (that adapts to the joint distribution between the response and covariates) that is obtained by taking the supremum of HSIC over a family of kernels, and show theoretically that *sup*-HSIC-SIS enjoys the sure screening property under some regularity conditions. We also propose two iterative versions of *sup*-HSIC-SIS that address issues inherent in any marginal screening procedure and are robust to the regularity assumptions. We show empirically that the proposed extensions along with *sup*-HSIC-SIS perform better than existing state-of-the-art approaches, while the theoretical analysis of these extensions are left out for future work.

A related RKHS based approach was previously proposed for feature selection in [99]. The approach uses HSIC metric and deals primarily with the low-dimensional setting (i.e., $n > p$) and is basically a model-free version of subset selection approaches used in linear regression settings. Comparing their empirical results with ours (see Sections 7.6.4 and 7.6.5), we note that while BA-HSIC is suitable for low-dimensions and to some extent for high-dimensional settings, it does not perform well in ultrahigh dimensional settings. We conjecture that BA-HSIC is inferior to DC-SIS and *sup*-HSIC-SIS in ultrahigh dimensional settings using arguments similar to the ones used in [68].

7.2 RKHS embedding of probabilities

Recently, the notion of embedding probability measures into a reproducing kernel Hilbert space (RKHS) has been proposed as a generalization to the classical kernel method (which embeds points from an input space into an RKHS) with a motivation

to provide a linear method for handling higher-order statistics of random variables [12, 95].

Formally, given a Borel probability measure, $\mathbb{P}$ defined on a topological space, $\mathcal{X}$, and the RKHS $(\mathcal{H}, k)$ of functions on $\mathcal{X}$ with bounded and measurable k as its reproducing kernel, the embedding of $\mathbb{P}$ into $\mathcal{H}$ is defined as $\mathbb{P}k := \int_{\mathcal{X}} k(\cdot, x)\, d\mathbb{P}(x)$. [48] defined the maximum mean discrepancy (MMD) RKHS distance between two Borel probability measures $\mathbb{P}$, $\mathbb{Q}$, as

$$\gamma_k(\mathbb{P}, \mathbb{Q}) \stackrel{\text{def}}{=} \left\| \int_{\mathcal{X}} k(\cdot, x)\, d\mathbb{P}(x) - \int_{\mathcal{X}} k(\cdot, x)\, d\mathbb{Q}(x) \right\|_{\mathcal{H}}.$$

When the kernel k is characteristic [102], the embeddings are injective, i.e., $\gamma_k(\mathbb{P}, \mathbb{Q}) = 0$ if and only if $\mathbb{P} = \mathbb{Q}$ and thus γ_k defines a metric on the space of probability measures.

One of the applications of the above metric is in capturing the degree of dependence between two random variables $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ with marginal distributions $\mathbb{P}(X)$ and $\mathbb{P}(Y)$ and jointly distributed as $\mathbb{P}(X, Y)$. Assuming $k : (\mathcal{X} \times \mathcal{Y})^2 \to \mathbb{R}$ is separable, i.e., $k((x, y), (x', y')) = k_{\mathcal{X}}(x, x')k_{\mathcal{Y}}(y, y')$, where $k_{\mathcal{X}} : \mathcal{X}^2 \to \mathbb{R}$ and $k_{\mathcal{Y}} : \mathcal{Y}^2 \to \mathbb{R}$ are reproducing kernels of $\mathcal{H}_{\mathcal{X}}$ and $\mathcal{H}_{\mathcal{Y}}$ respectively (so that $\mathcal{H} \cong \mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{Y}}$), γ_k^2 reduces to the Hilbert-Schmidt independence criterion [49] between X, Y, defined as

$$\begin{aligned} \gamma_k^2(\mathbb{P}(X, Y), \mathbb{P}(X)\mathbb{P}(Y)) &\stackrel{\text{def}}{=} \|\mathbb{P}(X, Y)k - \mathbb{P}(X)\mathbb{P}(Y)k\|_{\mathcal{H}}^2 \\ &= \mathsf{E}_{XX'YY'}\left[k_{\mathcal{X}}(X, X')k_{\mathcal{Y}}(Y, Y')\right] \\ &+ \mathsf{E}_{XX'}\left[k_{\mathcal{X}}(X, X')\right]\mathsf{E}_{YY'}\left[k_{\mathcal{Y}}(Y, Y')\right] \\ &- 2\mathsf{E}_{XY}\left[\mathsf{E}_{X'}[k_{\mathcal{X}}(X, X')]\mathsf{E}_{Y'}[k_{\mathcal{Y}}(Y, Y')]\right], \end{aligned} \tag{94}$$

where X' and Y' are independent copies of X and Y respectively. [49] showed that $\gamma_k(\mathbb{P}(X, Y), \mathbb{P}(X)\mathbb{P}(Y))$ is the Hilbert-Schmidt norm of the cross-covariance operator between $\mathcal{H}_{\mathcal{X}}$ and $\mathcal{H}_{\mathcal{Y}}$, with the property that when $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$ are characteristic, $\gamma_k(\mathbb{P}(X, Y), \mathbb{P}(X)\mathbb{P}(Y))$ is zero iff X and Y are independent. This crucial property

of γ_k will be exploited later in our screening framework. One difficulty is that this approach depends on tuning parameters associated with the kernel and selected in practice using heuristics. [101] proposed the following *sup*-HSIC variation:

$$\gamma(\mathbb{P}(X,Y),\mathbb{P}(X)\mathbb{P}(Y)) \stackrel{\text{def}}{=} \sup\{\gamma_k(\mathbb{P}(X,Y),\mathbb{P}(X)\mathbb{P}(Y)) : k \in \mathcal{K}\}.$$

Note that γ represents the maximal distance between $\mathbb{P}(X,Y)$ and $\mathbb{P}(X)\mathbb{P}(Y)$ over the family of kernels $\mathcal{K}$. If any $k \in \mathcal{K}$ is characteristic, then γ is a metric. Typical example includes the family of Gaussian kernels $\mathcal{K}_G$ when $k_{\mathcal{X}}(u,v) = k_{\mathcal{Y}}(u,v) \stackrel{\text{def}}{=} \{\exp^{-\sigma\|u-v\|_2^2} : \sigma \in \mathbb{R}_+\}$. See [101] for more details and examples.

In statistical problems, we are given n random samples $\{(X^{(1)},Y^{(1)}),\ldots,(X^{(n)},Y^{(n)})\}$ drawn i.i.d. from $\mathbb{P}(X,Y)$. Given these samples, an estimate $\widehat{\gamma}$ of *sup*-HSIC is defined as:

$$\begin{aligned}\widehat{\gamma}(\mathbb{P}(X,Y),\mathbb{P}(X)\mathbb{P}(Y)) &\stackrel{\text{def}}{=} \sup\{\|\mathbb{P}(X,Y)_n k - \mathbb{P}(X)_n\mathbb{P}(Y)_n k\|_{\mathcal{H}} : k \in \mathcal{K}\},\\ &= \frac{1}{n}\sup_{k_{\mathcal{X}}\in\mathcal{K}_{\mathcal{X}},k_{\mathcal{Y}}\in\mathcal{K}_{\mathcal{Y}}} \sqrt{\text{trace}\,(\mathbf{K}_{\mathcal{X}}\mathbf{H}\ \mathbf{K}_{\mathcal{Y}}\mathbf{H})}\end{aligned}$$

where $\mathbb{P}(X,Y)_n, \mathbb{P}(X)_n$ and $\mathbb{P}(Y)_n$ represent the empirical measures over the given samples. Above, $\mathbf{K}_{\mathcal{X}}$ and $\mathbf{K}_{\mathcal{Y}}$ are the $n \times n$ Gram matrices associated with $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$ respectively, and $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ where $\mathbf{I}$ is the $n \times n$ identity matrix ($\mathbf{1}$ is a $n \times 1$ vector of ones).

7.3 Screening via RKHS embedding

In this section, we describe how the *sup*-HSIC measure of independence could be used for feature screening in ultrahigh dimensions. We assume a response $Y \in \mathbb{R}^q$ (note that this notation differs slightly from the rest of the thesis; this is done in order to avoid potential confusion between the kernel k and output dimension) and covariates $X \in \mathbb{R}^{d_n}$, with d_n growing with n and q fixed (for simplicity). The method applies as well to more general topologic spaces $\mathcal{X},\mathcal{Y}$. We use X_r to denote the r-component of X and $X_{\mathcal{S}}$ to denote the components of X indexed by the elements of the set $\mathcal{S}$. We

denote the n training set samples as $\{(X^{(1)}, Y^{(1)}), \ldots, (X^{(n)}, Y^{(n)})\}$ where n can be very small compared to d_n. Under such an assumption, it is natural to assume that only a subset of covariates are related to the response Y.

Following [68], we define the set of relevant variables $\mathcal{M}$ and irrelevant variables $\mathcal{I}$ as:

$$\mathcal{M} = \{r : \mathbb{P}(Y|X) \text{ depends on } X_r\}$$
$$\mathcal{I} = \{r : \mathbb{P}(Y|X) \text{ does not depend on } X_r\}$$

where $\mathbb{P}(Y|X)$ is the conditional distribution of Y given X. Note that given $X_{\mathcal{M}}$, $X_{\mathcal{I}}$ is conditionally independent of Y and hence redundant while calculating the response. Using the above definitions, feature selection reduces to estimating the set $\mathcal{M}$ from a set of n iid samples.

A natural idea is to rank the covariates according to their degree of dependence to the response. In order to measure such a degree of dependence of the dimension X_r to Y, we use the *sup*-HSIC measure introduced in the previous section. Specifically, we use the *sup*-HSIC between the joint random variable (X_r, Y) and the marginals X_r and Y. Denoting the joint distribution of the vector (X_r, Y) as $\mathbb{P}(X_r, Y)$ and the marginal distribution of the dimensions X_r and Y as $\mathbb{P}(X_r)$ and $\mathbb{P}(Y)$ respectively, we define

$$\omega_r \stackrel{\text{def}}{=} \gamma_r(\mathbb{P}(X_r, Y), \mathbb{P}(X_r)\mathbb{P}(Y))$$

to be the measure of dependence between the r-component X_r and the response Y. Note that the greater γ_r is, the greater the degree of dependence and $\gamma_r = 0$ iff X_r is independent of Y. These properties make *sup*-HSIC suitable for ranking the dimensions of X according to the degree of dependence to the response Y. In practice, given n samples, we use the empirical estimator $\widehat{\omega}_r = \widehat{\gamma}_r(\mathbb{P}(X_r, Y)_n, \mathbb{P}(X_r)_n\mathbb{P}(Y)_n$ (defined in the previous section).

In order to select the relevant variables and estimate $\mathcal{M}$, we first compute $\widehat{\omega}_r$ for $r = 1, \ldots, d_n$ and define

$$\widehat{\mathcal{M}} = \{r : \widehat{\omega}_r \geq cn^{-\kappa}, \text{ for } 1 \leq r \leq d_n\}$$

where $0 \leq \kappa < 1/2$, as the estimated set of relevant features. Note that the set of relevant features is defined as the set of all dimensions that have dependence with the response greater than $cn^{-\kappa}$. The threshold defined here depends on n and when n is large variables with weaker dependence may be detected.

The approach above has several nice properties. First, the method is model free as it does not assume a specific regression model between X and Y. Second, the response Y may be a vector or more generally a graph or a ranking. As a result, the method can be used for feature selection in the case of multi-label classification and multivariate output regression. Third, the method chooses the kernel k in a principled way by selecting k from a family of positive definite kernels that maximizes the Hilbert Schmidt norm of the covariance operator. Finally, as we show in the next section the method generalizes the recently proposed state-of-the-art DC-SIS [68].

7.3.1 DC-SIS as a special case of *sup*-HSIC-SIS

In order to see how the proposed method generalizes the recent approach by [68], we appeal to the general equivalence between distance based independence metrics and kernel based independence metrics, as established by [90]. To summarize DC-SIS briefly, [68] uses distance covariance metric [104] as a measure of independence in the screening approach. In order to see the connection, we first need the following definition due to [74].

Definition 6. Let $(\mathcal{X}, \rho_{\mathcal{X}})$ and $(\mathcal{Y}, \rho_{\mathcal{Y}})$ be semi-metric spaces of negative type, with random variables X and Y taking values in $\mathcal{X}$ and $\mathcal{Y}$ respectively. The distance

covariance between X and Y is defined as

$$\begin{aligned}\mathrm{dcov}^2(X,Y) = &\mathsf{E}_{X,Y}\mathsf{E}_{X',Y'}\rho_{\mathcal{X}}(X,X')\rho_{\mathcal{Y}}(Y,Y') \\ &+ \mathsf{E}_X\mathsf{E}_{X'}\rho_{\mathcal{X}}(X,X')\mathsf{E}_Y,\mathsf{E}_{Y'}\rho_{\mathcal{X}}(X,X') \\ &- 2\mathsf{E}_{X,Y}\left(\mathsf{E}_{X'}\rho_{\mathcal{X}}(X,X')\mathsf{E}_{Y'}\rho_{\mathcal{X}}(X,X')\right).\end{aligned}$$

When $\mathcal{X} = \mathbb{R}^s$ and $\mathcal{Y} = \mathbb{R}^t$ with $\rho_{\mathcal{X}}(u,v) = \rho_{\mathcal{Y}}(u,v) = \|u-v\|$, dcov reduces to the distance used in [104]. The following result due to [91] establishes the equivalence between dcov and γ_k.

Theorem 7.3.1. *Let $(\mathcal{X}, \rho_{\mathcal{X}})$ and $(\mathcal{Y}, \rho_{\mathcal{Y}})$ be semi-metric spaces of negative type with $X \sim \mathbb{P}(X)$ and $Y \sim \mathbb{P}(Y)$ having joint $\mathbb{P}(X,Y)$. Let $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$ be kernels on $\mathcal{X}$ and $\mathcal{Y}$ that generate the respective metrics and denote $k((x,y),(x',y')) = k_{\mathcal{X}}(x,x')k_{\mathcal{Y}}(y,y')$. Then* $\mathrm{dcov}^2(X,Y) = 4\gamma_k^2(\mathbb{P}(X,Y), \mathbb{P}(X)\mathbb{P}(Y))$.

Example 11 in [91] shows that $k_q(x,x') = \frac{1}{2}(\|x\|^q + \|x'\|^q - \|x-x'\|^q)$, $x, x' \in \mathbb{R}^d$, $0 < q \leq 2$ generates a semi-metric, $\rho_q(x,x') = \|x-x'\|^q$ of negative type. Choosing $k_{\mathcal{X}} = k_{\mathcal{Y}} = k_1$ yields the *dcov* metric as proposed in [104], which is used in DC-SIS.

Using the *sup*-HSIC dependence measure in *sup*-HSIC-SIS generalizes DC-SIS in that it allows choosing more general exponent q, possibly k_q with q outside of the range $0 < q \leq 2$. This provides a richer set of independence measures between random variables, which in turn facilitates a better model-free feature selection. In addition, as we show later on *sup*-HSIC-SIS achieves better empirical results than the DC-SIS method.

7.4 Theoretical analysis

In this section, we prove the sure screening property of *sup*-HSIC-SIS for $\mathcal{X} \subset \mathbb{R}^{d_n}$ and $\mathcal{Y} \subset \mathbb{R}^q$. Our analysis applies to a range of kernel families and does not impose any moment conditions on the variables X and Y. Furthermore, it provides a simpler proof under relaxed assumption compared to [68] that also applies to DC-SIS. For

simplicity, we assume a fixed q be fixed, but one could also analyze the dependency on q to determine the joint scaling of q and d_n with n. We allow the cardinality of the dependence set to scale with n, i.e., $|\mathcal{M}_n| = s_n$. We assume the following assumptions:

A1 $\sup\{k_{\mathcal{X}}(x,x) : k_{\mathcal{X}} \in \mathcal{K}_{\mathcal{X}}, x \in \mathcal{X}\} = A < \infty$

A2 $\sup\{k_{\mathcal{Y}}(y,y) : k_{\mathcal{Y}} \in \mathcal{K}_{\mathcal{Y}}, y \in \mathcal{Y}\} = A < \infty$

A3 $\min_{r \in \mathcal{M}} \omega_r \geq 2cn^{-\kappa}$ for some $c > 0$ and $\kappa \in [0, 1/2)$.

Assumption A3 requires that *sup*-HSIC measure corresponding to the relevant variables cannot be too small, which is similar to condition 3 of [39] as well as to conditions in various other related chapters that analyzed marginal screening approaches. The proof of sure screening property of *sup*-HSIC-SIS in Theorem 7.4.1, uses an intermediate result in Lemma 1, stated and proved below. We first start with the following definition.

Definition 7. Let $\mathcal{G}$ be a class of functions on $\mathcal{X} \times \mathcal{X}$ and $\{\rho_1, \ldots \rho_n\}$ be independent Rademacher random variables. The homogeneous Rademacher chaos process of order two with respect to $\{\rho_1, \ldots \rho_n\}$ is defined as $\{n^{-1} \sum_{i<j}^{n} \rho_i \rho_j g(x_i, x_j) : g \in \mathcal{G}\}$ for some $\{x_1, \ldots, x_n\} \subset \mathcal{X}$. The Rademacher chaos complexity of $\mathcal{G}$ is defined as

$$U_n(\mathcal{G}; \{x_i\}) \stackrel{\text{def}}{=} \mathsf{E}_{\rho} \sup_{g \in \mathcal{G}} \left| \frac{1}{n} \sum_{i<j}^{n} \rho_i \rho_j g(x_i, x_j) \right|.$$

Lemma 5. *Let $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$ be measurable kernels satisfying assumptions* **A1** *and* **A2**. *Then for any $1 \leq r \leq d_n$, with probability at least $1 - \delta$ over the choice of samples, $\{(X_r^{(i)}, Y^{(i)})\}$,*

$$|\widehat{\omega}_r - \omega_r| \leq \sqrt{\frac{8U_n(\mathcal{K}; \{(X_r^{(i)}, Y^{(i)})\})}{n}} + \sqrt{\frac{8AU_n(\mathcal{K}_{\mathcal{X}}; \{X_r^{(i)}\})}{n}} + \sqrt{\frac{8AU_n(\mathcal{K}_{\mathcal{Y}}; \{y^{(i)}\})}{n}} + \sqrt{\frac{162A^2}{n} \log \frac{6}{\delta}} + \frac{6A}{\sqrt{n}}.$$

Proof. The proof technique is similar to that of Theorem 7 in [101]. Consider

$$|\widehat{\omega}_r - \omega_r| = |\widehat{\gamma}_r(\mathbb{P}(X_r, Y), \mathbb{P}(X_r)\mathbb{P}(Y)) - \gamma_r(\mathbb{P}(X_r, Y), \mathbb{P}(X_r)\mathbb{P}(Y))|$$
$$\leq \sup_{k \in \mathcal{K}} \|\mathbb{P}(X_r, Y)_n k - \mathbb{P}(X_r, Y)k\|_{\mathcal{H}} + \sup_{k \in \mathcal{K}} \|\mathbb{P}(X_r)_n \mathbb{P}(Y)_n k - \mathbb{P}(X_r)\mathbb{P}(Y)k\|_{\mathcal{H}}$$

.

We now bound the terms $\theta := \sup_{k \in \mathcal{K}} \|\mathbb{P}(X_r, Y)_n k - \mathbb{P}(X_r, Y)k\|_{\mathcal{H}}$ and $\phi := \sup_{k \in \mathcal{K}} \|\mathbb{P}(X_r)_n \mathbb{P}(Y)_n k - \mathbb{P}(X_r)\mathbb{P}(Y)k\|_{\mathcal{H}}$. Since θ satisfies the bounded difference property, using McDiarmid's inequality gives that with probability at least $1 - \frac{\delta}{6}$ over the choice of $\{(X_r^{(i)}, Y^{(i)})\}_{i=1}^n$, we have

$$\theta \leq \mathsf{E} \sup_{k \in \mathcal{K}} \|\mathbb{P}(X_r, Y)_n k - \mathbb{P}(X_r, Y)k\|_{\mathcal{H}} + \sqrt{\frac{2A^2}{n} \log \frac{6}{\delta}}. \tag{95}$$

By invoking symmetrization for $\mathsf{E} \sup_{k \in \mathcal{K}} \|\mathbb{P}(X_r, Y)_n k - \mathbb{P}(X_r, Y)k\|_{\mathcal{H}}$, we have

$$\mathsf{E}\, \theta \leq 2\mathsf{E}\, \mathsf{E}_\rho \sup_{k \in \mathcal{K}} \left\| \frac{1}{n} \sum_{i=1}^n \rho_i k(., (X_r^{(i)}, Y^{(i)})) \right\|_{\mathcal{H}}, \tag{96}$$

where $\{\rho_i\}_{i=1}^n$ represent i.i.d. Rademacher random variables and E_ρ represents the expectation w.r.t. $\{\rho_i\}$ conditioned on $\{(X_r^{(i)}, Y^{(i)})\}$. Since $\mathsf{E}_\rho \sup_{k \in \mathcal{K}} \left\| \frac{1}{n} \sum_{i=1}^n \rho_i k(., (X_r^{(i)}, Y^{(i)})) \right\|_{\mathcal{H}}$ satisfies the bounded difference property, by McDiarmid's inequality, with probability at least $1 - \frac{\delta}{6}$ over the choice of the random samples of size n, we have

$$\mathsf{E}\, \mathsf{E}_\rho \sup_{k \in \mathcal{K}} \left\| \frac{1}{n} \sum_{i=1}^n \rho_i k(., (X_r^{(i)}, Y^{(i)})) \right\|_{\mathcal{H}} \leq \sqrt{\frac{2A^2}{n} \log \frac{6}{\delta}} + \mathsf{E}_\rho \sup_{k \in \mathcal{K}} \left\| \frac{1}{n} \sum_{i=1}^n \rho_i k(., (X_r^{(i)}, Y^{(i)})) \right\|_{\mathcal{H}}. \tag{97}$$

By writing

$$\left\| \frac{1}{n} \sum_{i=1}^n \rho_i k(., (X_r^{(i)}, Y^{(i)})) \right\|_{\mathcal{H}} \leq \frac{A}{\sqrt{n}} + \frac{\sqrt{2}}{n} \sqrt{\left| \sum_{i<j}^n \rho_i \rho_j k((X_r^{(i)}, Y^{(i)}), (X_r^{(j)}, Y^{(j)})) \right|} \tag{98}$$

we have with probability at least $1 - \frac{\delta}{6}$, the following holds:

$$\mathsf{E}\, \mathsf{E}_\rho \sup_{k \in \mathcal{K}} \left\| \frac{1}{n} \sum_{i=1}^n \rho_i k(., (X_r^{(i)}, Y^{(i)})) \right\|_{\mathcal{H}} \leq \sqrt{\frac{2A^2}{n} \log \frac{6}{\delta}} + \frac{A}{\sqrt{n}} + \sqrt{\frac{2U_n(\mathcal{K}; \{(X_r^{(i)}, Y^{(i)})\})}{n}}. \tag{99}$$

Tying (95)-(99), we have that w.p. at least $1 - \frac{\delta}{3}$ over the choice of $\{(X_r^{(i)}, Y^{(i)})\}$, the following holds:

$$\theta \leq \sqrt{\frac{8U_n(\mathcal{K}; \{(X_r^{(i)}, Y^{(i)})\})}{n}} + \frac{2A}{\sqrt{n}} + \sqrt{\frac{18A^2}{n} \log \frac{6}{\delta}}. \tag{100}$$

Now we consider bounding ϕ

$$\begin{aligned}
\phi &\stackrel{\text{def}}{=} \sup_{k \in \mathcal{K}} \|\mathbb{P}(X_r)_n \mathbb{P}(Y)_n k - \mathbb{P}(X_r)\mathbb{P}(Y)k\|_{\mathcal{H}} \\
&= \sup_{k \in \mathcal{K}} \left\| \int k_{\mathcal{X}}(\cdot, x) \otimes k_{\mathcal{Y}}(\cdot, y)\, d[(\mathbb{P}(X_r) \times \mathbb{P}(Y) - (\mathbb{P}(X_r)_n \times \mathbb{P}(Y)_n)](x, y) \right\|_{\mathcal{H}} \\
&= \sup_{k \in \mathcal{K}} \left\| \int k_{\mathcal{X}}(\cdot, x)\, d\mathbb{P}(X_r)(x) \otimes \int k_{\mathcal{Y}}(\cdot, y)\, d\mathbb{P}(Y)(y) - \right. \\
&\qquad \left. \int k_{\mathcal{X}}(\cdot, x)\, d\mathbb{P}(X_r)_n(x) \otimes \int k_{\mathcal{Y}}(\cdot, y)\, d\mathbb{P}(Y)_n(y) \right\|_{\mathcal{H}} \\
&\leq \sup_{k \in \mathcal{K}} \left\| \int k_{\mathcal{X}}(\cdot, x)\, d(\mathbb{P}(X_r) - \mathbb{P}(X_r)_n)(x) \otimes \int k_{\mathcal{Y}}(\cdot, y)\, d\mathbb{P}(Y)(y) \right\|_{\mathcal{H}} \\
&\qquad + \sup_{k \in \mathcal{K}} \left\| \int k_{\mathcal{X}}(\cdot, x)\, d\mathbb{P}(X_r)_n(x) \otimes \int k_{\mathcal{Y}}(\cdot, y)\, d(\mathbb{P}(Y) - \mathbb{P}(Y)_n)(y) \right\|_{\mathcal{H}} \\
&= \sup_{k \in \mathcal{K}} \left\| \int k_{\mathcal{X}}(\cdot, x)\, d(\mathbb{P}(X_r) - \mathbb{P}(X_r)_n)(x) \right\|_{\mathcal{H}_{\mathcal{X}}} \left\| \int k_{\mathcal{Y}}(\cdot, y)\, d\mathbb{P}(Y)(y) \right\|_{\mathcal{H}_{\mathcal{Y}}} \\
&\qquad + \sup_{k \in \mathcal{K}} \left\| \int k_{\mathcal{X}}(\cdot, x)\, d\mathbb{P}(X_r)_n(x) \right\|_{\mathcal{H}_{\mathcal{X}}} \left\| \int k_{\mathcal{Y}}(\cdot, y)\, d(\mathbb{P}(Y) - \mathbb{P}(Y)_n)(y) \right\|_{\mathcal{H}_{\mathcal{Y}}} \\
&= \sup_{k_{\mathcal{X}} \in \mathcal{K}_{\mathcal{X}}} \left\| \int k_{\mathcal{X}}(\cdot, x)\, d(\mathbb{P}(X_r) - \mathbb{P}(X_r)_n)(x) \right\|_{\mathcal{H}_{\mathcal{X}}} \sup_{k_{\mathcal{Y}} \in \mathcal{K}_{\mathcal{Y}}} \left\| \int k_{\mathcal{Y}}(\cdot, y)\, d\mathbb{P}(Y)(y) \right\|_{\mathcal{H}_{\mathcal{Y}}} \\
&\qquad + \sup_{k_{\mathcal{Y}} \in \mathcal{K}_{\mathcal{Y}}} \left\| \int k_{\mathcal{Y}}(\cdot, y)\, d(\mathbb{P}(Y) - \mathbb{P}(Y)_n)(y) \right\|_{\mathcal{H}_{\mathcal{Y}}} \sup_{k_{\mathcal{X}} \in \mathcal{K}_{\mathcal{X}}} \left\| \int k_{\mathcal{X}}(\cdot, x)\, d\mathbb{P}(X_r)_n(x) \right\|_{\mathcal{H}_{\mathcal{X}}} \\
&\leq \sqrt{A} \sup_{k_{\mathcal{X}} \in \mathcal{K}_{\mathcal{X}}} \left\| \int k_{\mathcal{X}}(\cdot, x)\, d(\mathbb{P}(X_r) - \mathbb{P}(X_r)_n)(x) \right\|_{\mathcal{H}_{\mathcal{X}}} + \\
&\qquad \sqrt{A} \sup_{k_{\mathcal{Y}} \in \mathcal{K}_{\mathcal{Y}}} \left\| \int k_{\mathcal{Y}}(\cdot, y)\, d(\mathbb{P}(Y) - \mathbb{P}(Y)_n)(y) \right\|_{\mathcal{H}_{\mathcal{Y}}}.
\end{aligned}$$

Now,

$$\phi_{\mathcal{X}} \stackrel{\text{def}}{=} \sup_{k_{\mathcal{X}} \in \mathcal{K}_{\mathcal{X}}} \| \int k_{\mathcal{X}}(\cdot, x)\, d(\mathbb{P}(X_r) - \mathbb{P}(X_r)_n)(x) \|_{\mathcal{H}_{\mathcal{X}}}$$

and

$$\phi_{\mathcal{Y}} \stackrel{\text{def}}{=} \sup_{k_{\mathcal{Y}} \in \mathcal{K}_{\mathcal{Y}}} \| \int k_{\mathcal{Y}}(\cdot, y)\, d(\mathbb{P}(Y) - \mathbb{P}(Y)_n)(y) \|_{\mathcal{H}_{\mathcal{Y}}}$$

can be bounded by using Theorem 7 of [101], which yields that probability at least $1 - \frac{\delta}{3}$

$$\phi_{\mathcal{X}} \leq \sqrt{\frac{8U_n(\mathcal{K}_{\mathcal{X}}; \{X_r^{(i)}\})}{n}} + \frac{2\sqrt{A}}{\sqrt{n}} + \sqrt{\frac{18A}{n} \log \frac{6}{\delta}} \tag{101}$$

and

$$\phi_{\mathcal{Y}} \leq \sqrt{\frac{8U_n(\mathcal{K}_{\mathcal{Y}}; \{Y^{(i)}\})}{n}} + \frac{2\sqrt{A}}{\sqrt{n}} + \sqrt{\frac{18A}{n} \log \frac{6}{\delta}}. \tag{102}$$

Using (101) and (102), with probability at least $1 - \frac{2\delta}{3}$ over the choice of $\{X_r^{(i)}\}$ and $\{Y^{(i)}\}$, we have

$$\phi \leq \sqrt{\frac{8AU_n(\mathcal{K}_{\mathcal{Y}}; \{Y^{(i)}\})}{n}} + \frac{4A}{\sqrt{n}} + \sqrt{\frac{72A^2}{n} \log \frac{6}{\delta}} + \sqrt{\frac{8AU_n(\mathcal{K}_{\mathcal{X}}; \{X_r^{(i)}\})}{n}}. \tag{103}$$

Combining (100) and (103) provides the result. □

The above lemma will be helpful to prove the sure-screening property of the proposed approach in Theorem below.

Theorem 7.4.1. *Let $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$ be measurable kernels satisfying assumptions* **A1** *and* **A2**. *Define $D := \{(X^{(i)}, Y^{(i)})\}_{i=1}^n$. Then we have*

$$\begin{aligned}(\mathbb{P}(X,Y))^n \left(D \in (\mathcal{X} \times \mathcal{Y})^n : \max_{1 \leq r \leq d_n} |\widehat{\omega}_r - \omega_r| \geq cn^{-\kappa} \right) \\ \leq 6d_n \exp\left(- \frac{\left(cn^{\frac{1}{2}-k} - \mathcal{R}_n - 6A \right)^2}{162A^2} \right),\end{aligned} \tag{104}$$

where

$$\mathcal{R}_n \stackrel{\text{def}}{=} \sqrt{8AU_n(\mathcal{K}_{\mathcal{Y}}; \{Y^{(i)}\})} + \sup_r \left(\sqrt{8U_n(\mathcal{K}; \{(X_r^{(i)}, Y^{(i)})\})} + \sqrt{8AU_n(\mathcal{K}_{\mathcal{X}}; \{X_r^{(i)}\})} \right).$$

Furthermore if assumption **A3** *is also satisfied, then we have the following sure screening property:*

$$(\mathbb{P}(X,Y))^n \left(\mathcal{M} \subseteq \widehat{\mathcal{M}} \right) \geq 1 - O\left(s_n e^{- \frac{\left(cn^{\frac{1}{2}-k} - \mathcal{R}_n - 6A \right)^2}{162A^2}} \right).$$

Proof. The proof of (104) follows from applying Lemma 1 to each r followed by a union bound. In order to prove the sure screening property, if $\mathcal{M} \nsubseteq \widehat{\mathcal{M}}$, then there must exist some $r \in \mathcal{M}$ such that $\widehat{\omega}_r < cn^{-\kappa}$. But, from the assumption **A3**, we have that $|\widehat{\omega}_r - \omega_r| > cn^{-\kappa}$ for some $r \in \mathcal{M}$. Hence we have $\{\mathcal{M} \nsubseteq \widehat{\mathcal{M}}\} \subseteq \{|\widehat{\omega}_r - \omega_r| > cn^{-\kappa}\}$, for some $r \in \mathcal{M}$. Define $\Gamma = \{\max_{r\in\mathcal{M}} |\widehat{\omega}_r - \omega_r| \leq cn^{-\kappa}\}$. Then we have $\Gamma \subset \{\mathcal{M} \subseteq \widehat{\mathcal{M}}\}$ and we have $(\mathbb{P}(X,Y))^n \left(\mathcal{M} \subseteq \widehat{\mathcal{M}}\right) \geq (\mathbb{P}(X,Y))^n(\Gamma)$ and the following sequence of inequality holds

$$\begin{aligned}
\Pr(\Gamma) =& 1 - \Pr(\Gamma^c) = 1 - \Pr\left(\min_{r\in\mathcal{M}} |\widehat{\omega}_r - \omega_r| \geq cn^{-\kappa}\right) \\
=& 1 - s_n \Pr\left(|\widehat{\omega}_r - \omega_r| \geq cn^{-\kappa}\right) \\
\geq& 1 - O\left(s_n \exp\left(-\frac{\left(cn^{-\kappa+1/2} - \mathcal{R}_n - 6A\right)^2}{162A^2}\right)\right).
\end{aligned}$$

where $\Pr \stackrel{\text{def}}{=} (\mathbb{P}(X,Y))^n$. This completes the proof. □

Note that an important quantity controlling the rates is the term $\mathcal{R}_n$ that involves the Rademacher chaos complexities of $\mathcal{K}$, $\mathcal{K}_{\mathcal{X}}$ and $\mathcal{K}_{\mathcal{Y}}$. [101] has shown that for VC-subgraph classes of kernels, the Rademacher chaos complexity is bounded above by a constant that depends on the VC dimension of the class. Examples of such kernel classes in a d-dimensional Euclidean space include Gaussian, Laplacian, Matern class etc. We refer the reader to [101] for a detailed discussion and several more examples. In our setting, if $\mathcal{K}$, $\mathcal{K}_{\mathcal{X}}$ and $\mathcal{K}_{\mathcal{Y}}$ VC subgraph classes, then $\Pr\left(\max_{1\leq r\leq d_n} |\widehat{\omega}_r - \omega_r| \geq cn^{-\kappa}\right) \leq O\left(d_n \exp(-c_1 n^{1-2\kappa})\right)$ from which we observe that the proposed approach enables us to handle the ultrahigh dimensionality of $\log d_n = o(n^{1-2\kappa})$.

In order to control the false positive rates, if we assume that $\max_{r\notin\mathcal{M}} |\omega_r| = O(n^{-\kappa})$, then with probability tending to 1, we have

$$\max_{r\notin\mathcal{M}} |\widehat{\omega}_r| \leq C(n^{-\kappa}).$$

for some constant $C > 0$. By applying Theorem 7.4.1, we have: $\Pr(\mathcal{M} = \widehat{\mathcal{M}}) = 1 - O(1)$. This gives a model selection consistency result under the assumption that there is a strict separation between the set of relevant and irrelevant variables.

We analyze below the cardinality of the set $\widehat{\mathcal{M}}$.

7.4.1 Upper bounding the cardinality of $\widehat{\mathcal{M}}$

A main reason for performing feature screening is to reduce the dimensionality from exponential to something that can be handled such as polynomial in n (sample size). In this section, we show that by appropriately selecting the bounded on the kernel, one could make the cardinality of the estimated set grow polynomially in the sample size. Specifically, we have the following theorem.

Theorem 7.4.2. *Let $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$ be measurable kernels satisfying assumptions* **A1** *and* **A2**. *Then there exists a constant $c > 0$ such that,*

$$(\mathbb{P}(X,Y))^n \left(|\widehat{\mathcal{M}}| \leq O(n^{\kappa} d_n A) \right) \geq 1 - d_n e^{-\frac{\left(cn^{\frac{1}{2}-k} - \mathcal{R}_n - 6A \right)^2}{162A^2}}.$$

Proof. First we note that $\sum_{r=1}^{d_n} \omega_r \leq d_n \max_r \omega_r, \leq CAd_n = O(Ad_n)$. Now this would imply that $|\{r : \omega_r > \epsilon n^{-\kappa}\}|$ cannot exceed $O(n^{\kappa} A d_n)$ for any $\epsilon > 0$. Thus on the set, $\Upsilon = \{\max_{1 \leq r \leq d} |\widehat{\omega}_r - \omega_r| \leq \epsilon n^{-\kappa}\}$, $|\{r : \widehat{\omega}_r > 2\epsilon n^{-\kappa}\}|$ cannot exceed $|\{r : \omega_r > \epsilon n^{-\kappa}\}|$, which would be bounded by $O(n^{\kappa} A d_n)$. If we take $\epsilon = c/2$, we have $\Pr(|\widehat{\mathcal{M}}| \leq O(n^{\kappa} A p)) \geq \Pr(\Upsilon)$ and the conclusion follows from (104). □

The main consequence of the above theorem is that when $A = O(n^{\tau}/d_n)$, for some $\tau > 0$, then we have $|\widehat{\mathcal{M}}| = O(n^{\kappa+\tau})$ and thus the size of the selected set is of polynomial order in n. Compared to the initial case when the dimensionality is of exponential order, this is a huge improvement in terms of feature selection. This also gives us some insights on how to design or select kernels such that we can control the cardinality of the selected feature set size.

7.5 Iterative Screening procedures

Any screening method based on marginal computations suffers from the following problems [42]: (1) any irrelevant covariate that is highly correlated with the set of relevant covariates can be selected and (2) a marginally uncorrelated covariate that is jointly correlated with the response might not be selected. We propose below two approaches for handling such cases.

7.5.1 Method 1

We first consider the situation when important covariates are jointly correlated to the response but only weakly correlated marginally. In order to deal with this situation, we propose the following iterative method:

1. Compute *sup*-HSIC between each dimension and response and select the covariates that have $\omega_r > \lambda_t$. Let $\widehat{\mathcal{M}}_{(t)}$ be the set of selected covariates at round t with $X_{\widehat{\mathcal{M}}_{(t)}}$ being the set of selected features.

2. Compute *sup*-HSIC between $(Y, (X_{\widehat{\mathcal{M}}_{(t)}}, X_j))$ and marginal Y and $(X_{\widehat{\mathcal{M}}_{(t)}}, X_j))$ for all $j \in \widehat{\mathcal{M}}^c_{(t)}$. The selected feature set $\widehat{\mathcal{M}}'_{(t)}$ consists of covariates j for which the above calculated *sup*-HSIC is greater than the *sup*-HSIC between $(Y, X_{\widehat{\mathcal{M}}_{(t)}})$ and the marginal Y and $X_{\widehat{\mathcal{M}}_{(t)}}$. Update $\widehat{\mathcal{M}}_{(t)} = \widehat{\mathcal{M}}_{(t-1)} \cup \widehat{\mathcal{M}}'_{(t)}$

3. Repeat the procedure untill $\widehat{\mathcal{M}}_{(t)} = \widehat{\mathcal{M}}_{(t-1)}$ or until $|\bigcup_t \widehat{\mathcal{M}}_{(t)}| > n$.

In the above iterative approach, the threshold λ_t is set at a high value during the initial rounds and reduced as the rounds progress. In practice, it could be selected using cross-validation. Heuristics for selecting the threshold for such iterative methods could be found in [42]. The above iterative approach would be able to detect covariates that are marginally uncorrelated with the response (and hence not selected in initial rounds), but are jointly correlated because we measure *sup*-HSIC between the joint vector $(X_{\widehat{\mathcal{M}}_{(t)}}, X_j)$ and the response Y.

7.5.2 Method 2

This approach is motivated by the iterative screening procedure proposed by [39] which was based on residuals computed between the covariates and response under a linear model assumption. It is not possible to directly adopt such a procedure in our case, as the proposed approach is model-free.

We define $\mathbf{X}_{\widehat{\mathcal{M}}_{(t)}} \in \mathbb{R}^{n \times |\widehat{\mathcal{M}}_{(t)}|}$ to be the data matrix associated with selected covariates at round t and and $\mathbf{X}_{\widehat{\mathcal{M}}^c_{(t)}} \in \mathbb{R}^{n \times (p - |\widehat{\mathcal{M}}_{(t)}|)}$ to be the data matrix corresponding to the remaining covariates. We define the input residual matrix as the projection of complement of the selected variables in a particular step onto the orthogonal complement space of the selected variables in that step:

$$\mathbf{X}_r^{(t)} = \{\mathbf{I}_{n \times n} - \mathbf{X}_{\widehat{\mathcal{M}}_{(t)}} (\mathbf{X}_{\widehat{\mathcal{M}}_{(t)}}^\top \mathbf{X}_{\widehat{\mathcal{M}}_{(t)}})^{-1} \mathbf{X}_{\widehat{\mathcal{M}}_{(t)}}^\top \} \mathbf{X}_{\widehat{\mathcal{M}}^c_{(t)}}.$$

The key idea of this approach is that the input residual matrix at a particular step is uncorrelated with the space of selected variables in that step. Therefore covariates that would have been selected because they are correlated with a true relevant covariate (and hence correlated with the response) could be avoided in this approach. This leads to the following approach.

1. Calculate *sup*-HSIC for the original data set and let $\widehat{\mathcal{M}}_{(t)}$ be set of selected features at round t.

2. Compute the residual data matrix, $\mathbf{X}_r^{(t)} = \{\mathbf{I}_{n \times n} - \mathbf{X}_{\widehat{\mathcal{M}}_{(t)}} (\mathbf{X}_{\widehat{\mathcal{M}}_{(t)}}^\top \mathbf{X}_{\widehat{\mathcal{M}}_{(t)}})^{-1} \mathbf{X}_{\widehat{\mathcal{M}}_{(t)}}^\top \} \mathbf{X}_{\widehat{\mathcal{M}}^c_{(t)}}$ and compute *sup*-HSIC between $\mathbf{X}_r^{(t)}$ and the response to obtain the selected feature set $\widehat{\mathcal{M}}'_{(t)}$ and update $\widehat{\mathcal{M}}_{(t)} = \widehat{\mathcal{M}}_{(t-1)} \cup \widehat{\mathcal{M}}'_{(t)}$. Stop when $\widehat{\mathcal{M}}_{(t)} = \widehat{\mathcal{M}}_{(t-1)}$ or $|\bigcup_t \widehat{\mathcal{M}}_{(t)}| > n$.

As in the case of Method 1, the threshold for the initial round is set at a high value and subsequently lowered. Since the residual matrix at each step is not correlated with the selected covariates, the covariates that are strongly correlated with any of true

active covariates would not be selected. Also covariates that were actually correlated to the response (but were not selected) would now be detected easily.

7.6 Experiments

In this section, we report experimental results on various synthetic and real-world data sets. These experiments demonstrate the advantage of the proposed approach (*sup*-HSIC-SIS) over various feature screening approaches. For the experiments on synthetic data, we consider the data settings from [68] in order to make a direct comparison to their approach (as DC-SIS is the current state-of-the-art). For evaluation on real-world data, we consider a very high dimensional gene data set and a multi-label data set. In both cases our proposed approach performs significantly better than the existing approaches.

7.6.1 Synthetic data – univariate response

We generated the following synthetic data: $X \sim N(0, \Sigma)$ where $\Sigma \in \mathbb{R}^{d \times d}$ with entries $\sigma_{i,j} = 0.8^{|i-j|}$. We set $n = 200$ and $p = 5000$. We generate the response Y according to three models

1. $Y = c_1\beta_1 X_1 X_2 + c_3\beta_2 1(X_{12} < 0) + c_4\beta_3 X_{22} + \epsilon$
2. $Y = c_1\beta_1 X_1 X_2 + c_3\beta_2 1(X_{12} < 0) X_{22} + \epsilon$
3. $Y = c_1\beta_1 X_1 + c_2\beta_2 X_2 + c_3\beta_3 1(X_{12} < 0) + \exp(c_4|X_2 2||)\epsilon$

where $\beta_j = (-1)^U(a + |Z|)$ where $a = 4\log n/\sqrt{n}$, $U \sim$ Bernoulli(0.4) and $Z, \epsilon \sim N(0.1)$. Note that all models are non-linear in X_{12} and the third model is heteroscedastic.

We also generated a fourth data set, where the relationship between the response and covariates is given by the following joint model for each r: $\mathbb{P}(X_r, Y) \propto 1 + \sin(lx)\sin(ly)$ where the support is $[-\pi, \pi] \times [-\pi, \pi]$ and l is an integer. Note that

when $l = 0$, X_r and Y are independent and as $|l|$ increases they become dependent wherein the joint distribution departs from the uniform at higher frequencies, making it hard to detect from small sample sizes. We set $l = 10$ for $r = 1, 2, 3, 4$ and $l = 0$ for the remaining values of r (the response is dependent on the first four covariates only).

We compared the following approaches:

- HSIC-SIS with $k_q(z, z') = 1/2 \left(\|z\|^q + \|z'\|^q - \|z - z'\|^q \right)$ at $q = 1, 0.5, 0.25$
- *sup*-HSIC-SIS with $\mathcal{K} = \{k_q : 0 < q \leq 2\}$(Note that $q = 1$ corresponds to DC-SIS)
- *sup*-HSIC-SIS with a Gaussian kernel
- non-parametric independence screening (NIS) of [37].

Table 16 shows $P(\mathcal{M}^* \subset \widehat{\mathcal{M}})$ and $P(\mathcal{M}^* = \widehat{\mathcal{M}})$ computed over 500 experiments. Note that the proposed *sup*-HSIC-SIS approach performs better than other approaches. In some cases the Gaussian kernel performs better, while in other cases the distance kernel performs better. The fourth model in particular clearly demonstrates the advantage of the proposed approach (the other approaches are not able to detect the specific type of dependency). Selecting a kernel for a given task is a more involved problem which we hope to address in the future (a simple step in this direction would be to consider a convex combination of base kernels).

7.6.2 Synthetic data – multivariate response

In this experiment, we deal with multivariate outputs. We generate X as before and generate Y from a normal distribution with mean zero and conditional covariance matrix $\Sigma_{Y|X}$ given by $\sigma_{11} = \sigma_{22} = 1$ and $\sigma_{12} = \sigma_{21} = \sigma(X)$. We consider two correlation functions for $\sigma(X)$ given by

1. $\sigma(X) = \sin(\beta_1^\top X)$ where $\beta_1 = (0.8, 0.6, 0, \ldots, 0)$

Table 16: Probability of support recovery using the distance kernel and Gaussian kernel: First four rows correspond to $P(\mathcal{M}^* = \widehat{\mathcal{M}})$ (corresponding to models 1, 2, 3 and 4 respectively) and the last four rows correspond to $P(\mathcal{M}^* \subset \widehat{\mathcal{M}})$. The very last row corresponds to the average cardinality of selected set.

NIS	$q = 1$	$q = \frac{1}{2}$	$q = \frac{1}{4}$	$\sup_q k_q$	Gauss.
$P(\mathcal{M}^* = \widehat{\mathcal{M}})$					
0.78	0.79	0.82	0.84	0.88	0.87
0.73	0.75	0.79	0.80	0.83	0.84
0.73	0.73	0.75	0.78	0.82	0.82
0.35	0.40	0.52	0.60	0.71	0.80
$P(\mathcal{M}^* \subset \widehat{\mathcal{M}})$					
0.96	0.98	1.00	1.00	1.00	1.00
0.94	0.95	0.99	1.00	1.00	1.00
0.93	0.96	1.00	1.00	1.00	1.00
0.6	0.69	0 .72	0.75	0.92	0.98
$\lvert\widehat{\mathcal{M}}\rvert$					
10.1	7.4	5.4	4.4	4.2	4.2

2. $\sigma(X) = \{\exp(\beta_2^\top X - 1)/\exp(\beta_2^\top X + 1)\}$ where $\beta_2 = (2 - U_1, 2 - U_2, 2 - U_3, 2 - U_4, 0, \ldots, 0)$ with U_i drawn i.i.d. from Uniform$[0, 1]$.

Note that for this experiment, the NIS method could not be used directly as it cannot handle multivariate outputs. Hence, we only compared our approach to DC-SIS, whose results are presented in Table 17. It is clear from Table 17 that *sup*-HSIC-SIS performs better in this setup as well.

7.6.3 Synthetic data – Iterative screening

In this section, we examine the performance of the iterative screening procedures (see Section 7.5). The experiments show that this method performs better than *sup*-HSIC-SIS (using a Gaussian kernel). We use the synthetic data described by [39], which consists of a linear model $y = \beta^\top x + \epsilon$ with $\beta \in \mathbb{R}^p$ and $\epsilon \sim N(0, 1)$. We set $\beta = (5, 5, 5, -15\sqrt{\rho}, 0, \ldots, 0)$ with $p = 2000$ and we draw $n = 100$ covariates x from a mean zero normal distribution with $\Sigma_{d\times d} = \sigma_{ij}$, with entries $\sigma_{ii} = 1$ for $i = 1, \ldots, p$ and $\sigma_{i4} = \sigma_{4i} = \sqrt{\rho}$ for $i \neq 4$ and $\sigma_{ij} = \rho$ for $i \neq j$, $i \neq 4$ and $j \neq 4$. Note that all

Table 17: Probability of support recovery using the distance kernel and Gaussian kernel. First two rows correspond to $P(\mathcal{M}^* = \widehat{\mathcal{M}})$ and the last three rows correspond to $P(\mathcal{M}^* \subset \widehat{\mathcal{M}})$. The very last row corresponds to the average cardinality of selected set over all experiments.

$q = 1$	$q = \frac{1}{2}$	$q = \frac{1}{4}$	$\sup_q k_q$	Gaussian
$P(\mathcal{M}^* = \widehat{\mathcal{M}})$				
0.79	0.85	0.86	0.91	0.90
0.77	0.81	0.85	0.87	0.89
$P(\mathcal{M}^* \subset \widehat{\mathcal{M}})$				
0.97	0.99	1.00	1.00	1.00
0.96	0.97	0.98	1.00	1.00
$\lvert\widehat{\mathcal{M}}\rvert$				
9.4	6.7	5.2	4.3	4.4

Table 18: Advantage of iterative methods over sup-HSIC-SIS. The values reported are estimates of $P(\mathcal{M}^* \subset \widehat{\mathcal{M}})$ over 1000 trials.

ρ	0	0.1	0.5	0.9
sup-HSIC-SIS	0.98	0.89	0.54	0.42
Method 1	1.00	1.00	0.99	0.95
method 2	1.00	1.00	1.00	1.00

predictors except x_4 are equally correlated with correlation coefficient ρ. In addition, x_4 has correlation coefficient ρ with all other predictors and is independent of y, but x_4 belongs to the active set when $\rho \neq 0$ (we vary ρ among the set $\{0, 0.1, 0.5, 0.9\}$).

Both iterative algorithms completed 2 iterations before attaining the stopping criterion (the threshold parameter was set by cross-validation). We repeat the experiment for 1000 trials and report the probability of including all correct variables in the estimated set $P(\mathcal{M}^* \subset \widehat{\mathcal{M}})$ (see Table 18).

As Table 18 shows the non-iterative version performed poorly, while both iterative algorithms performed well. Method 1 performed slightly worse compared to Method 2 because it has to deal with multivariate *sup*-HSIC evaluations in the second step, which is relatively hard to do with less samples.

7.6.4 Gene array data set

We analyze the performance of the feature selection methods on the Affymetric GeneChip Rat Genome Array data set. The dataset, previously used in [89] and [56], consists of 120 rat subjects from which $18,975$ different probes sets (genes) from eye tissue were measured. Following [56], the intensity values were normalized and gene expression levels were analyzed on a logarithmic scale. Specifically, we are interested in finding the genes that are most related to TRIM32 gene, the reason being that this gene was recently found to cause Bardet-Biedl syndrome, a topic of interest in the biological community. The data set is highly challenging with $n = 120$, $p = 18,975$, and a non-linear relationship between the covariates and the response.

We used *sup*-HSIC-SIS with Gaussian kernel to select the important genes and compared it to BA-HSIC, NIS and DC-SIS methods. BA-HSIC cannot actually handle high dimensionality because of its design; we just use it for comparison purpose. For the experiment, we used 100 training samples to select the features (genes), and fitted an additive model (with functions in Sobolev classes) using the selected features, and compared the predictive error (PE) on the remaining 20 points. BA-HSIC performs poorly in this regime (small n, large d) and fails to select many important genes that are selected by all the other methods (in addition to exhibiting worse predictive accuracy). Both NIS and DC-SIS select 8 genes, whereas the proposed approach selects 7 genes. The predictive accuracy of the proposed approach is smaller implying that maybe the additional gene selected by the other methods is not actually necessary to explain the response, a potential advantage to biologists.

7.6.5 Multi-label classification data set

In our final experiment, we evaluate the performance of *sup*-HSIC-SIS (using Gaussian kernel), DC-SIS and BA-HSIC on 4 different yahoo multi-label data sets: arts, business, education and health [114]. The task is to select features first using the above

Table 19: Gene data set: Cardinality of selected set and predictive error (PE) under an additive model.

Method	Cardinality	PE
BA-HSIC	12.32	4.32
NIS	7.73	0.47
DC-SIS	7.21	0.45
sup-HSIC-SIS	6.76	0.39

Table 20: Test set classification error on the multi-label data sets. The number in the bracket correspond to the cardinality of selected feature set.

Data set	BA-HSIC	DC-SIS	Proposed
Arts	(967) 25.87	(658) 14.32	(435) 9.54
Business	(1231) 26.32	(743) 15.64	(611) 10.11
Edu	(1123) 21.02	(643) 11.31	(533) 9.21
Health	(1045) 22.54	(764) 13.42	(564) 10.74

three methods and perform classification in the next step using one-vs-all multi-label SVM approach. For each of the data sets, the number of samples was set at $n = 1000$ (with balanced label proportions). The dimensionality of (X, Y) for the data sets are $(17973, 19), (16621, 17), (20782, 14), (18430, 14)$ respectively. Table 20 shows the classification accuracy and the cardinality of the selected features for different data sets. The proposed approach achieves better classification accuracy with a fewer number of features.

CHAPTER VIII

CONCLUSION

The aim of this chapter is to summarize and reiterate the contributions made in this thesis. We also discuss possible extensions of the proposed approaches. Furthermore, continuing along the lines of the thesis, we outline concrete problems that are similar in sprit, to the main theme of the thesis.

8.1 Summary and Key Contributions

In Chapters 2 and 3, we developed a novel framework for estimating margin-based risks using only unlabeled data. We show that it performs well in practice on several different data sets. 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 $\mathbb{P}(Y)$ 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 $n \to \infty$ without any labeled data.

The results in Chapter 2 are applicable only to additive (with both linear and non-linear components) classifiers, which form an extremely important class of classifiers especially in the high dimensional case. In the non-linear classification scenario, it is worth examining if the CLT assumptions on the mapped high-dimensional feature space could be used for building non-linear classifiers via the kernel trick. 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 our framework be extended to semi-supervised learning where a few labels do exist? Can it be extended 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.

In Chapter 5, We proposed a two-step estimation procedure based on a specialized random effects model for dealing with joint sparsity regularization and demonstrated its advantage over the group-Lasso formulation. The proposed approach highlights the fact that enforcing interesting structure on covariance of the coefficients is better for obtaining joint sparsity in the coefficients. Future work also includes (i) relaxing the assumptions made in the theoretical analysis, (ii) exploring more complex models like imposing group-mean structure on the parameters for additional flexibility, (iii) other additive decomposition of the covariance matrix with complementary regularizers and (iv) using locally-smoothed covariance estimates for time-varying joint sparsity.

We proposed a framework for multi-output prediction based on parsimonious modeling on the output space in Chapter 4. By selecting a subset of the output dimensions (landmarks) and focusing on modeling the dependency of that subset of y on x, we reduce the sample complexity considerably. This is most noticeable when the output dimensionality is high and the different component feature high correlation. Our experiments indicate that the proposed method outperforms standard multi-output methods in both the classification and regression scenarios.

In Chapter 6, we proposed a simple framework for incorporating similarity in feature space and space or time into sparse coding. We also propose in this chapter modifying sparse coding by replacing the lasso optimization stage by marginal regression and adding a constraint to enforce incoherent dictionaries. The resulting

algorithm is significantly faster (speedup of about two-orders of magnitude over standard sparse coding). This facilitates scaling up the sparse coding framework to large dictionaries, an area which is usually restricted due to intractable computation.

This work leads to several interesting follow-up directions. On the theoretical side: (i) local convergence of proposed approach is interesting to analyze and (ii) it is also interesting to explore tighter generalization error bounds by directly analyzing the solutions of the marginal regression iterative algorithm. Methodologically, it is interesting to explore: (i) using an adaptive or non-constant kernel bandwidth to get higher accuracy, and (iv) alternative incoherence constraints that may lead to easier optimization and scaling up.

We proposed an RKHS embedding approach for feature screening of ultrahigh dimensional data in Chapter 7. The proposed approach is model-free and works with multivariate and general output spaces such as graphs or rankings. We prove the feature screening consistency of the proposed approach and empirically demonstrated its capability in handling ultrahigh dimensional regimes on various synthetic and real-world data sets. Furthermore, we proposed two iterative screening methods to counter some problems exhibited by the marginal screening based feature selection approaches.

8.2 Related open problems

We conclude the thesis by outlining two concrete open questions that arise naturally from the work described in this thesis.

8.2.1 Joint Regularization for Multiple Low-rank Estimation

Group-Lasso has been introduced in the vector regression context as a way of exploiting shared structure in multiple vector regression situation. One could examine if a similar extension of the approach proposed in Chapter 5, could be used in the context of trace-regression, where we group singular values at the same level of several

matrices to enforce joint low-rank structure. In order to analyze this particular procedure, one would require obtaining novel tail bounds beyond than the ones existing in random matrix theory.

8.2.2 Sparse-additive Near-separable Nonnegative Matrix Factorization

Non-negative matrix factorization could be seen as a method of learning undercomplete features of a data sample, that are non-negative.The proposed landmark selection approach, in Chapter 4. could essentially be used for provably computing non-negative matrix factorization under near-separability assumption. An interesting investigation is to develop a non-parametric version of non-negative matrix factorization under sparse additivity and near-separability assumption for extracting non-linear features from the data set.

REFERENCES

[1] Argyriou, A., Evgeniou, T., and Pontil, M., "Convex multi-task feature learning," *Machine Learning*, vol. 73, no. 3, pp. 243–272, 2008.

[2] Argyriou., A., Micchelli, C. A., Pontil, M., and Ying, Y., "A spectral regularization framework for multi-task structure learning," *NIPS*, 2008.

[3] Balasubramanian, K., Donmez, P., and Lebanon, G., "Unsupervised supervised learning II: Training margin based classifiers without labels," *Journal of Machine Learning Research*, 2011.

[4] Balasubramanian, K. and Lebanon, G., "The landmark selection method for multiple output prediction," *Proc. of the International Conference on Machine Learning*, 2013.

[5] Balasubramanian, K., Sriperumbudur, B. K., and Lebanon, G., "Ultrahigh dimensional feature screening via rkhs embeddings," *International Conference on Artificial Intelligence and Statistics*, 2013.

[6] Balasubramanian, K., Yu, K., and Lebanon, G., "Smooth sparse coding via marginal regression for learning sparse representations," in *Proceedings of The 30th International Conference on Machine Learning*, 2013.

[7] Balasubramanian, K., Yu, K., and Zhang, T., "High-dimensional joint sparsity random effects model for multi-task learning," *Uncertainty in Artificial Intelligence*, 2013.

[8] Bartlett, P., Bousquet, O., and Mendelson, S., "Local rademacher complexities," *The Annals of Statistics*, 2005.

[9] Behboodian, J., "Information matrix for a mixture of two normal distributions," *Journal of statistical computation and simulation*, vol. 1, no. 4, pp. 295–314, 1972.

[10] Bengio, S., Pereira, F., Singer, Y., and Strelow, D., "Group sparse coding," *NIPS*, vol. 22, 2009.

[11] 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.

[12] Berlinet, A. and Thomas-Agnan, C., *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. London, UK: Kluwer Academic Publishers, 2004.

[13] Bertsekas, D., "On the goldstein-levitin-polyak gradient projection method," *IEEE Transactions on Automatic Control*, 1976.

[14] Bi, W. and Kwok, J., "Multi-label classification on tree and dag structured hierarchies," *ICML*, 2011.

[15] Bigot, J., Biscay, R., Loubes, J. M., and Alvarez, L. M., "Group Lasso estimation of high-dimensional covariance matrices," *Journal of Machine Learning Research*, 2011.

[16] Bigot, J., Biscay, R., Loubes, J. M., and Muñiz-Alvarez, L., "Nonparametric estimation of covariance functions by model selection," *EJS*, vol. 4, 2010.

[17] Bishop, Y., Fienberg, S., and Holland, P., *Discrete multivariate analysis: theory and practice*. MIT press, 1975.

[18] Blitzer, J., Dredze, M., and Pereira, F., "Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification," in *Proc. of ACL '07*, 2007.

[19] Breiman, L., "Bias, variance, and arcing classifiers," Tech. Rep. 460, Statistics department, University of California, 1996.

[20] Breiman, L. and Friedman, J. H., "Predicting multivariate responses in multiple linear regression," *Journal of the Royal Statistical Society:B*, vol. 59, 1997.

[21] Bronstein, A., Sprechmann, P., and Sapiro, G., "Learning efficient structured sparse models," *ICML*, 2012.

[22] Candes, E. and Tao, T., "The Dantzig selector: Statistical estimation when p is much larger than n," *Annals of Statistics*, vol. 35, no. 6, pp. 2313–2351, 2007.

[23] Candes, E., Li, X., Ma, Y., and Wright, J., "Robust principal component analysis?," *JACM*, 2011.

[24] Castelli, V. and Cover, T. M., "On the exponential value of labeled samples," *Pattern Recognition Letters*, vol. 16, no. 1, pp. 105–111, 1995.

[25] 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.

[26] Cesa-Bianchi, N., Gentile, C., and Zaniboni, L., "Incremental algorithms for hierarchical classification," *Journal of Machine Learning Research*, 2006.

[27] Chandrasekaran, V., Sanghavi, S., Parrilo, P., and Willsky, A., "Rank-sparsity incoherence for matrix decomposition," *SIAM Journal of Optimization*, 2011.

[28] Cover, T. M. and Thomas, J. A., *Elements of Information Theory*. John Wiley & Sons, second ed., 2005.

[29] Cox, D., Little, J., and O'Shea, D., *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer, 2006.

[30] Dai, W., Yang, Q., Xue, G.-R., and Yu, Y., "Boosting for transfer learning," in *Proc. of International Conference on Machine Learning*, 2007.

[31] Davidson, J., *Stochastic limit theory: An introduction for econometricians*. Oxford University Press, USA, 1994.

[32] Devroye, L. and Lugosi, G., *Combinatorial methods in density estimation*. 2001.

[33] Donmez, P., Lebanon, G., and Balasubramanian, K., "Unsupervised supervised learning I: Estimating classification and regression error rates without labels," *Journal of Machine Learning Research*, vol. 11, no. April, pp. 1323–1351, 2010.

[34] Duda, R. O., Hart, P. E., and Stork, D. G., *Pattern classification*. Wiley, 2001.

[35] Efron, B. and Tibshirani, R. J., *An Introduction to the Bootstrap*. Chapman & Hall, 1997.

[36] Fan, J., Fan, Y., and Lv, J., "High dimensional covariance matrix estimation using a factor model," *Journal of Econometrics*, vol. 147, no. 1, pp. 186–197, 2008.

[37] Fan, J., Feng, Y., and Song, R., "Nonparametric independence screening in sparse ultra-high-dimensional additive models," *J. Amer. Statist. Assoc.*, vol. 106, no. 494, pp. 544–557, 2011.

[38] Fan, J. and Li, R., "Variable selection via nonconcave penalized likelihood and its oracle properties," *J. Amer. Statist. Assoc.*, vol. 96, no. 456, pp. 1348–1360, 2001.

[39] Fan, J. and Lv, J., "Sure independence screening for ultrahigh dimensional feature space," *J. Roy. Statist. Soc. Ser. B*, vol. 70, pp. 849–911, 2008.

[40] Fan, J. and Lv, J., "Sure independence screening for ultrahigh dimensional feature space," *JRSS: B(Statistical Methodology)*, 2008.

[41] Fan, J. and Lv, J., "A selective overview of variable selection in high dimensional feature space," *Statistica Sinica*, vol. 20, pp. 101–148, 2010.

[42] Fan, J., Samworth, R., and Wu, Y., "Ultrahigh dimensional feature selection: Beyond the linear model," *J. of Machine Learning Research*, vol. 10, pp. 2013–2038, 2009.

[43] Ferguson, T. S., *A Course in Large Sample Theory*. Chapman & Hall, 1996.

[44] Frank, A. and Asuncion, A., "UCI machine learning repository," *University of California, School of Information and Computer Science, Irvine, CA. Available at http://archive.ics.uci.edu/ml/*, 2010.

[45] Genovese, C. R., Jin, J., Wasserman, L., and Yao, Z., "A comparison of the lasso and marginal regression," *JMLR*, 2012.

[46] Gomes, R., Krause, A., and Perona, P., "Discriminative clustering by regularized information maximization," in *Advances in Neural Information Processing Systems 24*, 2010.

[47] Gong, P., Ye, J., and Zhang, C., "Multi-stage multi-task feature learning," *NIPS*, 2012.

[48] Gretton, A., Borgwardt, K. M., Rasch, M., Schölkopf, B., and Smola, A., "A kernel method for the two sample problem," in *Advances in Neural Information Processing Systems 19*, pp. 513–520, MIT Press, 2007.

[49] Gretton, A., Bousquet, O., Smola, A., and Schölkopf, B., "Measuring statistical dependence with Hilbert-Schmidt norms," in *Proc. of the 16th International Conference on Algorithmic Learning Theory*, pp. 63–77, 2005.

[50] Hand, D. J., "Recent advances in error rate estimation," *Pattern Recognition Letters*, vol. 4, no. 5, pp. 335–346, 1986.

[51] Harville, D., "Maximum likelihood approaches to variance component estimation and to related problems," *JASA*, pp. 320–338, 1977.

[52] Hastie, T. and Loader, C., "Local regression: Automatic kernel carpentry," *Statistical Science*, pp. 120–129, 1993.

[53] Hoeffding, W. and Robbins, H., "The central limit theorem for dependent random variables," *Duke Mathematical Journal*, vol. 15, pp. 773–780, 1948.

[54] Hsu, D., Kakade, S. M., Langford, J., and Zhang, T., "Multi-label prediction via compressed sensing," *NIPS*, 2009.

[55] Hsu, D., Kakade, S. M., and Zhang, T., "Robust Matrix Decomposition with Outliers," *IEEE Transactions on Information Theory*, 2011.

[56] Huang, J., Horowitz, J., and Wei, F., "Variable selection in nonparametric additive models," *Annals of statistics*, vol. 38, no. 4, pp. 2282–2313, 2010.

[57] Huang, J. and Zhang, T., "The benefit of group sparsity," *The Annals of Statistics*, vol. 38, no. 4, pp. 1978–2004, 2010.

[58] Izenman, A., "Reduced-rank regression for the multivariate linear model," *Journal of multivariate analysis*, vol. 5, 1975.

[59] J. Ratsaby, J. and Venkatesh, S. S., "Learning from a mixture of labeled and unlabeled examples with parametric side information," in *Annual conference on Computational learning theory*, 1995.

[60] Jenatton, R., Mairal, J., Obozinski, G., and Bach, F., "Proximal methods for sparse hierarchical dictionary learning," *ICML*, 2010.

[61] Ji, P. and Jin, J., "UPS delivers optimal phase diagram in high-dimensional variable selection," *Annals of Statistics*, vol. 40, no. 1, pp. 73–103, 2012.

[62] Joachims, T., "Making large-scale svm learning practical," in *Advances in Kernel Methods - Support Vector Learning* (Schölkopf, B., Burges, C., and Smola, A., eds.), MIT Press, 1999.

[63] Kim, M., Kumar, S., Pavlovic, V., and Rowley, H., "Face tracking and recognition with visual constraints in real-world videos," in *CVPR*, 2008.

[64] Kläser, A., Marszałek, M., and Schmid, C., "A spatio-temporal descriptor based on 3d-gradients," in *BMVC*, 2008.

[65] Lang, K., "Newsweeder: Learning to filter netnews," in *International Conference on Machine Learning*, 1995.

[66] Lee, H., Battle, A., Raina, R., and Ng, A., "Efficient sparse coding algorithms," in *NIPS*, 2007.

[67] 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.

[68] Li, R., Zhong, W., and Zhu, L., "Feature screening via distance correlation learning," *J. Amer. Statist. Assoc.*, vol. 107, pp. 1129–1139, 2012.

[69] Liu, H., Lafferty, J., and Wasserman, L., "Nonparametric regression and classification with joint sparsity constraints," *NIPS*, 2008.

[70] Liu, J., Luo, J., and Shah, M., "Recognizing realistic actions from videos in the wild," in *CVPR*, 2009.

[71] Loader, C., *Local regression and likelihood*. Springer Verlag, 1999.

[72] Lounici, K., Pontil, M., Tsybakov, A. B., and van der Geer, S., "Taking advantage of sparsity in multi-task learning," *COLT09*, 2009.

[73] Lowe, D. G., "Object recognition from local scale-invariant features," *CVPR*, 1999.

[74] Lyons, R., "Distance covariance in metric spaces," *Annals of Probability*, 2012. To appear.

[75] Magnus, J. R. and Neudecker, H., *Matrix differential calculus with applications in statistics and econometrics*. Wiley, 1988.

[76] Maurer, A. and Pontil, M., "K-dimensional coding schemes in hilbert spaces," *Information Theory, IEEE Transactions on*, vol. 56, no. 11, pp. 5839–5846, 2010.

[77] Meier, L. and Bühlmann, P., "Smoothing l1-penalized estimators for high-dimensional time-course data," *Electronic Journal of Statistics*, 2007.

[78] Obozinski, G., Wainwright, M. J., and Jordan, M. I., "support union recovery in high-dimensional multivariate regression," *Annals of statistics*, vol. 39, 2011.

[79] Papoulis, A., *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1984.

[80] 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.

[81] Quadrianto, N., Smola, A. J., Caetano, T. S., and Le, Q. V., "Estimating labels from label proportions," *Journal of Machine Learning Research*, vol. 10, pp. 2349–2374, 2009.

[82] Raina, R., Battle, A., Lee, H., Packer, B., and Ng, A., "Self-taught learning: transfer learning from unlabeled data," in *ICML*, 2007.

[83] Ramirez, I., Lecumberry, F., and Sapiro, G., "Sparse modeling with universal priors and learned incoherent dictionaries," *Tech Report, IMA, University of Minnesota*, 2009.

[84] Ravikumar, P., Lafferty, J., Liu, H., and Wasserman, L., "Sparse additive models," *J. Roy. Statist. Soc. Ser. B*, vol. 71, pp. 1009–1030, 2009.

[85] Reinsel, G. C. and Velu, R. P., *Multivariate reduced-rank regression: theory and applications*. Springer New York, 1998.

[86] Rifkin, R. and Klautau, A., "In defense of one-vs-all classification," *The Journal of Machine Learning Research*, vol. 5, pp. 101–141, 2004.

[87] Rinott, Y., "On normal approximation rates for certain sums of dependent random variables," *Journal of Computational and Applied Mathematics*, vol. 55, no. 2, pp. 135–143, 1994.

[88] Rohde, A. and Tsybakov, A. B., "Estimation of high-dimensional low-rank matrices," *The Annals of Statistics*, vol. 39, 2011.

[89] Scheetz, T., Kim, K., Swiderski, R., Philp, A., Braun, T., Knudtson, K., Dorrance, A., DiBona, G., Huang, J., Casavant, T., Sheffield, V., and Stone, E., "Regulation of gene expression in the mammalian eye and its relevance to eye disease," *Proceedings of the National Academy of Sciences*, vol. 103, no. 39, pp. 14429–14434, 2006.

[90] Sejdinovic, D., Gretton, A., Sriperumbudur, B., and Fukumizu, K., "Hypothesis testing using pairwise distances and associated kernels," pp. 1111–1118, 2012.

[91] Sejdinovic, D., Sriperumbudur, B., Gretton, A., and Fukumizu, K., "Equivalence of distance-based and RKHS-based statistics in hypothesis testing," *arXiv:1207.6076*, 2012.

[92] 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 Internation Conference on Knowledge Discovery and Data Mining*, pp. 614–622, 2008.

[93] Sigg, C. D., Dikk, T., and Buhmann, J. M., "Learning dictionaries with bounded self-coherence," *IEEE Transactions on Signal Processing*, 2012.

[94] Silva, J., Marques, J., and Lemos, J., "Selecting landmark points for sparse manifold learning," *NIPS*, vol. 18, 2006.

[95] Smola, A. J., Gretton, A., Song, L., and Schölkopf, B., "A Hilbert space embedding for distributions," in *Proc. 18th International Conference on Algorithmic Learning Theory*, pp. 13–31, Springer-Verlag, Berlin, Germany, 2007.

[96] 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.

[97] 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.

[98] Solodov, M. V., "Convergence analysis of perturbed feasible descent methods," *Journal of Optimization Theory and Applications*, 1997.

[99] Song, L., Smola, A., Gretton, A., Bedo, J., and Borgwardt, K., "Feature selection via dependence maximization," *J. of Machine Learning Research*, vol. 13, pp. 1393–1434, 2012.

[100] Sprechmann, P., Ramírez, I., Sapiro, G., and Eldar, Y. C., "C-hilasso: A collaborative hierarchical sparse modeling framework," *Signal Processing, IEEE Transactions on*, vol. 59, 2011.

[101] Sriperumbudur, B., Fukumizu, K., Gretton, A., Lanckriet, G., and Schölkopf, B., "Kernel choice and classifiability for RKHS embeddings of probability distributions," in *Advances in Neural Information Processing Systems 22*, pp. 1750–1758, MIT Press, 2009.

[102] Sriperumbudur, B. K., Gretton, A., Fukumizu, K., Schölkopf, B., and Lanckriet, G. R. G., "Hilbert space embeddings and metrics on probability measures," *J. of Machine Learning Research*, vol. 11, pp. 1517–1561, 2010.

[103] Sturmfels, B., *Solving Systems of Polynomial Equations*. American Mathematical Society, 2002.

[104] Székely, G., Rizzo, M., and Bakirov, N., "Measuring and testing dependence by correlation of distances," *Annals of Statistics*, vol. 35, no. 6, pp. 2769–2794, 2007.

[105] Tai, F. and Lin, H. T., "Multi-label classification with principle label space transformation," *Neural Computation*, 2012.

[106] Teicher, H., "Identifiability of finite mixtures," *The Annals of Mathematical Statistics*, vol. 34, no. 4, pp. 1265–1269, 1963.

[107] Tewari, A. and Bartlett, P. L., "On the consistency of multiclass classification methods," *Journal of Machine Learning Research*, pp. 1007–1025, 2007.

[108] Tibshirani, R., "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc. Ser. B*, vol. 58, pp. 267–288, 1996.

[109] Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K., "Sparsity and smoothness via the fused lasso," *JRSS:B*, vol. 67, 2005.

[110] Tropp, J. A., "Algorithms for simultaneous sparse approximation. part II: Convex relaxation," *Signal Processing*, vol. 86, pp. 589–602, 2006.

[111] Tropp, J., "Greed is good: Algorithmic results for sparse approximation," *Information Theory, IEEE*, 2004.

[112] Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y., "Large margin methods for structured and interdependent output variables," *Journal of Machine Learning Research*, 2006.

[113] Tsoumakas, G., I., I. K., and Vlahavas, "Mining multi-label data," *Data mining and knowledge discovery handbook*, pp. 667–685, 2010.

[114] Ueda, N. and Saito, K., "Parametric mixture models for multi-labeled text," in *Advances in Neural Information Processing Systems 15*, pp. 721–728, MIT Press, 2003.

[115] Vainsencher, D., Mannor, S., and Bruckstein, A., "The sample complexity of dictionary learning," *JMLR*, 2011.

[116] Vapnik, V. N., *The Nature of Statistical Learning Theory*. Springer, second ed., 2000.

[117] Vens, C., Struyf, J., Schietgat, L., Džeroski, S., and Blockeel, H., "Decision trees for hierarchical multi-label classification," *Machine Learning*, 2008.

[118] Wang, H., Ullah, M., Klaser, A., Laptev, I., and Schmid, C., "Evaluation of local spatio-temporal features for action recognition," in *BMVC*, 2009.

[119] Wipf, D. P. and Rao, B. D., "An empirical bayesian strategy for solving the simultaneous sparse approximation problem," *Signal Processing, IEEE Transactions on*, vol. 55, no. 7, pp. 3704–3716, 2007.

[120] Wright, S. J., Nowak, R. D., and Figueiredo, M. A. T., "Sparse reconstruction by separable approximation," *Signal Processing, IEEE Transactions on*, vol. 57, 2009.

[121] Yang, J., Yu, K., Gong, Y., and Huang, T., "Linear spatial pyramid matching using sparse coding for image classification," *Computer Vision and Pattern Recognition*, 2009.

[122] Yang, J., Yu, K., Gong, Y., and Huang, T., "Linear spatial pyramid matching using sparse coding for image classification," in *CVPR*, 2009.

[123] Yang, J., Yu, K., and Huang, T., "Supervised translation-invariant sparse coding," in *CVPR*, 2010.

[124] Yu, K., Lin, Y., and Lafferty, J., "Learning image representations from the pixel level via hierarchical sparse coding," in *CVPR*, 2011.

[125] Yu, K., Zhang, T., and Gong, Y., "Nonlinear learning using local coordinate coding," *NIPS*, 2009.

[126] Yuan, M. and Lin, Y., "Model selection and estimation in regression with grouped variables," *JRSS: Series B*, vol. 68, 2006.

[127] Zangwill, W., *Nonlinear programming: a unified approach*. Prentice Hall, 1969.

[128] Zhang, T., "Analysis of multi-stage convex relaxation for sparse regularization," *JMLR*, 2010.

[129] Zhao, P. and Yu, B., "On model selection consistency of lasso," *Journal of Machine Learning Research*, 2006.

[130] Zhao, P. and Yu, B., "On model selection consistency of lasso," *J. of Machine Learning Research*, vol. 7, pp. 2541–2563, 2007.

[131] Zhu, X. and Goldberg, A. B., *Introduction to semi-supervised learning.* Morgan & Claypool Publishers, 2009.

[132] Zou, H., Hastie, T., and Tibshirani, R., "Sparse principal component analysis," *Journal of computational and graphical statistics*, 2006.

VITA

Krishnakumar Balasubramanian is a Ph.D. candidate in the School of Computational Science and Engineering, College of Computing, Georgia Tech, where he also obtained his M.S. in 2010. Before joining Georgia Tech, he completed B.Eng. from Madras Institute of Technology, India, in 2008. He received the Facebook fellowship award for 2013-14 and the ICML best paper runner-up award in 2013. His primary research interests are in statistics and machine learning.