GENERALIZATION BOUNDS FOR COMPRESSED LEARNING WITH HARD SUPPORT VECTOR MACHINES, AND MULTICLASS LEARNING WITH ERROR CORRECTING OUTPUT CODES

A Dissertation

by

PAUL ROBERT MCVAY

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Krishna Narayanan
Co-Chair of Committee,	Tie Liu
Committee Members,	Robert Nevels
	Thomas Schlumprecht
Head of Department,	Miroslav M. Begovic

December 2020

Major Subject: Electrical Engineering

ABSTRACT

This dissertation addresses two problems important to machine learning: how does compressed data affect the performance of hard support vector machines (hard-SVM) and how does one efficiently create a multiclass classifier with good performance.

The first section provides a theoretical analysis to characterize when compressed learning, i.e., learning on compressed data, is possible with the separability assumption. Using these results, we give an upper bound on the compression ratio that maintains separability in the compressed domain. We provide results for the case when sparsity is assumed as well as when no sparsity assumption is made. Furthermore, we provide theoretical results to show how the generalization bound changes with respect to the compression ratio used. These results allow for theoretical justifications in choosing the best compression matrix given the particular design parameters at hand. Additionally, as required for the analysis presented, we extend the existing hard-SVM bounds to the case when a bias term is allowed.

The second section presents a novel output coding approach to multiclass classification. Our algorithm optimizes the encoder for a channel code based coding matrix to ensure the maximum minimum distance of the coding matrix. The optimization procedure uses the properties of the code to run extremely fast, $O(klogk)$. We demonstrate the need for the optimal minimum distance for the coding matrix by proving a generalization bound for both hard and soft decoding. These bounds beat the previously published tight asypmtotic growth rate with respect to k. Finally, we present empirical results to validate our approach.

ACKNOWLEDGMENTS

First, I would like to thank both of my wonderful advisors, Dr. Tie Liu and Dr. Krishna Narayanan for their support and guidance.

I would like to thank all of my committee members, Dr. Robert Nevels and Dr. Thomas Schlumprecht, for their time, suggestions, and feedback.

I would like to thank Dr. Simon Foucart and Dr. Grigoris Paouris for there teachings and help in understanding the theory of compression and random matrices.

Finally, I would like to thank all the friends and family who supported me emotionally through my graduate studies.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professor Tie Liu [advisor], Professor Krishna Narayanan [advisor], and Professor Robert Nevels of the Department of Electrical and Computer Engineering, and Professor Thomas Schlumprecht of the Department of Mathematics

Funding Sources

Graduate study was supported in part by the National Science Foundation and in part by the Eric D. Rubin Professorship endowment.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 Introduction

This dissertation explores two specialized cases of machine learning. Fundamentally, machine learning is the transformation of data into knowledge by a computer. More precisely, an algorithm chooses an output function from a function class based on a sequence of data points called a training set. Ideally, the output function selected will perform well on underlying unknown probability distribution that generated the training set.

The basic setup for a machine learning task is a domain space, $\mathcal{X}$, and a label space, $\mathcal{Y}$. It is assumed that there is an unknown distribution, μ, over the domain space and the label space, $\mathcal{X} \times \mathcal{Y}$. This distribution is unknown to the algorithm. The training set is a sequence of m points from $\mathcal{X} \times \mathcal{Y}$, i.e. $\mathcal{S} = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m))$. It is assumed that each point in the training set is generated independently according to μ. That is, $\mathcal{S} \sim \mu^{\otimes m}$.

The goal of a machine learning algorithm is to select an output function based on the training set that performs well with respect to the underlying distribution. A machine learning algorithm typically has a function class $\mathcal{H} : \mathcal{X} \to \mathcal{Y}$ from which the output function is selected.

Two main problems have been addressed in this dissertation:

1. (Compressed Data Hard-SVM) The first problem asks under what conditions does the linear separability assumption holds before and after compression. The condition is required to run hard-SVM on compressed data. This dissertation also addresses how the generalization bound changes when using compressed data.

2. (Multiclass Coding Matrix Design) The second problem asks how to optimally and efficiently design a coding matrix for a fast reduction of multiclass classification to binary classification.

1.1.1 Compressed Data Hard-SVM

The first problem addresses the use of hard-SVM on compressed data. Hard-SVM is a powerful and popular machine learning algorithm for binary classification that requires linear separability. However, there are several instances in which the user may wish to run hard-SVM on compressed data instead of the full dataset. These reasons include: 1) to reduce the computation time of the algorithm; 2) to reduce the storage cost of the algorithm; 3) to reduce the data transmission cost of server/node configurations.

Given the computational and data transmission needs of running hard-SVM on compressed data, it is important to understand the performance trade-off of using compressed data. In previous work, Calderbank et al. [4] analyzed the generalization performance after compression when using soft-SVM and compressing the data through techniques from compressed sensing.

In our work in chapters 4 and 5 we analyze the conditions under which hard-SVM can be run after compression with and without sparsity as well as the generalizational difference of performance in these setups.

We show that the linear separability assumption holds after compression in two scenarios. If we assume the data is sparse before compression, we show that the linear separability assumptions holds after compression if the $2s$-restricted isometric constant of the compression matrix Φ is in an acceptable range. This result is presented in chapter 4. Furthermore, if we lose the sparsity assumption, we can still show the linear separability assumption holds after compression if the compressed dimension m is in an acceptable range. This result is shown in chapter 5.

After showing the feasibility of hard-SVM after compression, we then show how the generalization bound relates before and after compression. We show that that generalization bound after compression scales the generalization bound before compression by a multiplicative factor.

These results are important for the machine learning community to understand. As the size of datasets continues to grow, techniques must be used to effectively deal with the size. We provide a theoretical perspective on the feasibility of hard-SVM after compression as well as a characterizing the performance loss allowing a machine learning practitioner to carefully consider the trade-offs

of using compression.

1.1.2 Multiclass Coding Matrix Design

The second problem addresses how to efficiently design a coding matrix for multiclass classification. Multiclass classification is commonly done by reducing the multiclass problem into a series of binary problems. This is done through the use of a coding matrix.

As the size of the datasets grow, the coding matrix must be designed efficiently, require minimal binary classifiers, and perform with good generalization to be practical and useful. It has previously been shown that creating the optimal coding matrix based on the training data is intractable [5].

We present an algorithm that uses the properties of channel codes to build a coding matrix with large minimum Hamming distance that is also optimized with respect to a similarity metric to create easy binary partitions.

Our theoretical results show that the minimum Hamming distance of the coding matrix is an important parameter in the generalization bounds of multiclass classification. These results are shown by studying the Rademacher complexity bounds and presented in chapter 6.

Our optimization procedure with respect to the similarity matrix is shown to run in $\mathcal{O}(k \log k)$ time where k is the number of classes. Previous coding matrix optimization schemes based on the similarity matrix have required iterations until convergence of steps with $\mathcal{O}(k^3)$ complexity. Furthermore, we show our coding matrix design outperforms previous methods on four datasets: ODP, ImageNet, SUN, AlOI.

Our results are important for several reasons. First, our theoretical results show the importance of the minimal Hamming distance of the in the generalization bounds. As full training set optimization of the coding matrix is intractable, practitioners need to know which broad coding matrix properties are important for generalization. Second, our optimization procedure based on the similarity matrix is extremely fast allowing it to be used on the large datasets like ODP and ImageNet where previous approaches are computationally infeasible. Additionally, our coding matrix design performs better than previous methods which is the main point of building a learning algorithm.

2. BACKGROUND

The simplest situation for a machine learning task is binary classification. Binary classification consists of a domain space, $\mathcal{X}$, and a label space, $\mathcal{Y}$, where the label space has two elements, i.e. $|\mathcal{Y}| = 2$. For simplicity, the two elements of the label space are denoted $\mathcal{Y} = \{-1, 1\}$. This leads to easy predictions for a function. A function f can be said to predict a label $\hat{y}$ for a domain point $\mathbf{x}$ as $\hat{y} = \text{sign}(f(\mathbf{x}))$.

For the theoretical analysis of machine learning, it is assumed that there is an unknown distribution, μ, over the domain space and the label space, $\mathcal{X} \times \mathcal{Y}$. This distribution is unknown to the algorithm. The training set is a sequence of m points from $\mathcal{X} \times \mathcal{Y}$, i.e. $\mathcal{S} = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m))$. It is assumed that each point in the training set is generated independently according to μ. That is, $\mathcal{S} \sim \mu^{\otimes m}$.

The common approach to selecting an output function, $h_{\mathcal{S}}$, from a function class, $\mathcal{H} : \mathcal{X} \to \mathbf{Y}$, for a learning algorithm, $\mathcal{A}$, is empirical risk minimization. Here, the output function, $h_{\mathcal{S}}$, depends on the particular training set, $\mathcal{S}$, received. A loss function, $l : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, is a measure of the performance of a function, h, for a particular domain point, $\mathbf{x}$ and label, y. Ideally, an algorithm would output a function $h_{\text{ideal}} \in \mathcal{H}$ such that the true risk, $\mathcal{L}_\mu(h) = \mathbb{E}_{(\mathbf{x},y)\sim\mu}[l(h, \mathbf{x}, y)]$ is minimized. That is, $h_{\text{ideal}} = \text{argmin}_{h\in\mathcal{H}}\mathcal{L}_\mu(h)$. However, as μ is unknown this cannot be computed. Therefore, the algorithm considers the training set a proxy for the distribution. The empirical risk is defined $\mathcal{L}_{\mathcal{S}}(h) = \frac{1}{m}\sum_{i=1}^{m} l(h, \mathbf{x}_i, y_i)$. The algorithm then selects a function that minimizes the empirical risk. That is, $h_{\mathcal{S}} = \text{argmin}_{h\in\mathcal{H}}\mathcal{L}_{\mathcal{S}}(h)$.

As the algorithm selects the output function based on the empirical risk but the user cares about the true risk, we want to bound the gap between the two values. That is, we want to be able to say theoretically that $\mathcal{L}_\mu(h_{\mathcal{S}}) \leq \mathcal{L}_{\mathcal{L}}(h_{\mathcal{S}}) + \epsilon_{\text{error}}$ where the error is sufficiently small. These results are commonly called generalization bounds and have two varieties: data-independent and data-dependent.

Data-Independent: One of the more famous data-independent generalization bounds for bi-

nary classification is based on Vapnik-Chervonenkis dimension's (VC-dimension) of the function class $\mathcal{H}$ [19, 18]. This bound is considered a *data-independent* bound as it doesn't depend on the particular training set. The VC-dimension, VCdim, of a function class, $\mathcal{H}$, is the maximal size of a set $C \subset \mathcal{X}$ that can be shattered by $\mathcal{H}$. The generalization bound is stated below:

Theorem 1 (Shalev-Shwartz)**.** *Let $\mathcal{H}$ be a function class with* $\text{VCdim}(\mathcal{H}) = d$*. Then, for every distribution μ, every $\delta \in (0,1)$, and every $h \in \mathcal{H}$, with probability of at least $1-\delta$ over the choice of $\mathcal{S} \sim \mu^{\otimes m}$,*

$$\mathcal{L}_\mu(h) \leq \mathcal{L}_\mathcal{S}(h) + \frac{4 + \sqrt{d \log(2em/d)}}{\delta\sqrt{2m}} \tag{2.1}$$

Data-Dependent: One example of a data-dependent generalization bound is based on Rademacher complexities. This bound is *data-dependent* as it depends on the particular training set received. The Rademacher complexity of a set $A \subset \mathbb{R}^m$ is defined $R(A) = \frac{1}{m}\mathbb{E}_\sigma[\sup_{a\in A} \sum_{i=1}^m \sigma_i a_i]$. If we define $l \circ \mathcal{H} \circ \mathcal{S} = \{(l(h, \mathbf{x}_1, y_1), \ldots, l(h, \mathbf{x}_m, y_m)) : h \in \mathcal{H}\}$ we have the following generalization bound from [18, 2, 14]:

Theorem 2 (Shalev-Shwartz)**.** *If $\forall \mathbf{x}, y \in \mathcal{X} \times \mathcal{Y}$ and $\forall h \in \mathcal{H}$ we have $|l(h, \mathbf{x}, y)| \leq c$, then with probability of at least $1-\delta$ for all $h \in \mathcal{H}$*

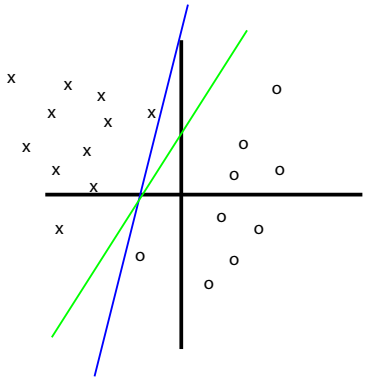
$$\mathcal{L}_\mu(h) \leq \mathcal{L}_\mathcal{S}(h) + 2R(l \circ \mathcal{H} \circ \mathcal{S}) + 4c\sqrt{\frac{2\ln(4/\delta)}{m}} \tag{2.2}$$

2.1 Support Vector Machines

Suppose we consider our current setup of empirical risk minimization with affine classes when the domain space is a euclidean space i.e., $\mathcal{X} = \mathbb{R}^d$. Our hypothesis class would be $\mathcal{H} = \{\text{sign}(\langle \mathbf{w}, \mathbf{x}\rangle + b) : \mathbf{w} \in \mathbb{R}^d,\ b \in \mathbb{R}\}$. If our loss function was the zero-one loss function ($l(h, \mathbf{x}, y) = \mathbb{1}_{h(\mathbf{x})\neq y}$), we would select as the output function

$$(\mathbf{w}_\mathcal{S}, b_\mathcal{S}) = \underset{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}}{\text{argmin}} \frac{1}{m}\sum_{i=1}^m \mathbb{1}_{\text{sign}(\langle \mathbf{w}, \mathbf{x}_i\rangle + b)\neq y_i}.$$

Figure 2.1: Multiple affine functions have zero empirical risk



However, as illustrated by Fig. 2.1, there can be several minimizing hyperplanes. Intuitively, we prefer the hyperplane that is farthest from the training sets (the green hyperplane). This relates to the concept of margins. Instead of looking at a function class $\mathcal{H} : \mathcal{X} \to \mathcal{Y}$ we look at a function class $\mathcal{F} : \mathcal{X} \to \mathbb{R}$. We suppose that an output function, $f \in \mathcal{F}$, gives a measure of the confidence in the prediction as well as a label prediction. Intuitively, if a domain point is close to the decision boundary of the labeling function, we would have less confidence in the prediction then if the point was far away from the decision boundary. The margin of the point is a signed distance relating to the confidence and correctness of point with respect to the labeling function. If we consider a function $f \in \mathcal{F}$, then we denote the label of $\mathbf{x}$ according to f as $\text{sign}(f(\mathbf{x}))$ and the confidence of the prediction of $\mathbf{x}$ according to f as $|f(\mathbf{x})|$. The margin, $m_f(\mathbf{x}, y)$, is the confidence, $|f(\mathbf{x})|$, if the label is correct i.e., $\text{sign}(f(\mathbf{x})) = y$ and the negative confidence, $-|f(\mathbf{x})|$, if the label is incorrect i.e., $\text{sign}(f(\mathbf{x})) \neq y$. We note that this if-then statement can be simplified to $m_f(\mathbf{x}, y) = yf(x)$.

The hard support vector machine (hard-SVM) algorithm, as opposed to empirical risk minimization, maximizes the minimum margin over the training set [18]. That is,

$$(\mathbf{w}_{\text{SVM}}, b_{\text{SVM}}) = \underset{\mathbf{w}\in\mathbb{R}^d, \|\mathbf{w}\|=1, b\in\mathbb{R}}{\text{argmax}} \min_{i\in[m]} y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b). \tag{2.3}$$

This optimization procedure can be solved as the equivalent quadratic optimization procedure below:

$$(\mathbf{w}_{\mathcal{S}}, b_{\mathcal{S}}) = \underset{(\mathbf{w},b)}{\text{argmin}} ||\mathbf{w}||^2 \tag{2.4}$$

$$\text{s.t.} \quad \forall i, \quad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 \tag{2.5}$$

where $\mathbf{w}_{\text{SVM}} = \frac{\mathbf{w}_{\mathcal{S}}}{\|\mathbf{w}_{\mathcal{S}}\|}$ and $b_{\text{SVM}} = \frac{b_{\mathcal{S}}}{\|\mathbf{w}_{\mathcal{S}}\|}$.

This chapter presents our results for the generalization of hard-SVM. In Shalev-Shwartz and Ben-David [18] and Kakade et al. [13], they present an analysis of the generalization of hard-SVM when restricting SVM to linear classes. That is, they ignore the bias term for simplicity. The SVM optimization procedure in this case in the following:

$$\mathbf{w}_{\mathcal{S}} = \underset{\mathbf{w}}{\operatorname{argmin}} ||\mathbf{w}||^2 \tag{3.1}$$

$$\text{s.t.} \quad \forall i, \quad y_i \langle \mathbf{w}, \mathbf{x}_i \rangle \geq 1 \tag{3.2}$$

We extend the analysis of the generalization of hard-SVM to the case presented in (2.4) and (2.5). To do this we first show a bound on the Rademacher complexity of affine classes.

3.1 Rademacher Complexity of Affine Class

Lemma 3. *Let $\mathcal{S}_x = (\mathbf{x_1}, \ldots, \mathbf{x_m})$ be vectors in a Hilbert space.*
Define $\mathcal{H} \circ \mathcal{S} = \{(\langle \mathbf{w}, \mathbf{x_1} \rangle + b, \ldots, \langle \mathbf{w}, \mathbf{x_m} \rangle + b) : \|\mathbf{w}\| \leq B_1, |b| \leq B_2\}$. Let $\mathcal{R}(A)$ denote the Rademacher complexity of A. Then,

$$\mathcal{R}(\mathcal{H} \circ \mathcal{S}) \leq \frac{B_1 max_i \|\mathbf{x_i}\| + B_2}{\sqrt{m}} \tag{3.3}$$

Proof:

$$m\mathcal{R}(\mathcal{H} \circ \mathcal{S}) = \mathbb{E}_\sigma \left[\sup_{\mathbf{a} \in \mathcal{H} \circ \mathcal{S}} \sum_{i=1}^{m} \sigma_i a_i \right]$$

$$= \mathbb{E}_\sigma \left[\sup_{\substack{\mathbf{w}: \|\mathbf{w} \leq B_1 \| \\ b: |b| \leq B_2}} \sum_{i=1}^{m} \sigma_i \left(\langle \mathbf{w}, \mathbf{x} \rangle + b \right) \right]$$

$$= \mathbb{E}_\sigma \left[\sup_{\mathbf{w}: \|\mathbf{w}\| \leq B_1} \sum_{i=1}^{m} \sigma_i \langle \mathbf{w}, \mathbf{x} \rangle + \sup_{b: |b| \leq B_2} \sum_{i=1}^{m} \sigma_i b \right]$$

$$= \mathbb{E}_\sigma \left[\sup_{\mathbf{w}:\|\mathbf{w}\| \leq B_1} \sum_{i=1}^{m} \sigma_i \langle \mathbf{w}, \mathbf{x} \rangle \right] + \mathbb{E}_\sigma \left[\sup_{b:|b| \leq B_2} b \cdot \sum_{i=1}^{m} \sigma_i \right]$$

By lemma 26.10 in Shalev-Shwartz and Ben-David [18] which was taken from [13]we can bound the left term by $B_1\sqrt{m} \cdot \max_i \|\mathbf{x_i}\|$ giving us

$$\leq B_1\sqrt{m} \cdot \max_i \|\mathbf{x_i}\| + \mathbb{E}_\sigma \left[\sup_{b:|b| \leq B_2} b \cdot \sum_{i=1}^{m} \sigma_i \right] \tag{3.4}$$

Now considering the right term

$$\sup_{b:|b| \leq B_2} b \cdot \sum_{i=1}^{m} \sigma_i = \begin{cases} -B_2 \cdot \sum_{i=1}^{m} \sigma_i & \sum_{i=1}^{m} \sigma_i < 0 \\ B_2 \cdot \sum_{i=1}^{m} \sigma_i & \sum_{i=1}^{m} \sigma_i \geq 0 \end{cases}$$

$$= \left| B_2 \sum_{i=1}^{m} \sigma_i \right|$$

Substituting this in

$$\mathbb{E}_\sigma \left[\left| B_2 \sum_{i=1}^{m} \sigma_i \right| \right] = \sum_{i=0}^{\lfloor \frac{m}{2} \rfloor} \binom{m}{i} \left(\frac{1}{2}\right)^m (-B_2)(2i-m) + \sum_{i=\lceil \frac{m}{2} \rceil}^{m} \binom{m}{i} \left(\frac{1}{2}\right)^m (B_2)(2i-m)$$

$$= 2 \sum_{i=0}^{\lfloor \frac{m}{2} \rfloor} \binom{m}{i} \left(\frac{1}{2}\right)^m (B_2)(m-2i)$$

$$= \left(\frac{1}{2}\right)^{m-1} B_2 \sum_{i=0}^{\lfloor \frac{m}{2} \rfloor} \binom{m}{i} (m-2i)$$

$$= \left(\frac{1}{2}\right)^{m-1} B_2 \left(m \sum_{i=0}^{\lfloor \frac{m}{2} \rfloor} \binom{m}{i} - 2 \sum_{i=0}^{\lfloor \frac{m}{2} \rfloor} \binom{m}{i} i \right) \tag{3.5}$$

Using lemma 5 and lemma 6 from Appendix 3.2, this becomes.

$$= \left(\frac{1}{2}\right)^{m-1} B_2 \left(m \begin{cases} 2^{m-1} & m \text{ is odd} \\ 2^{m-1} + \binom{m}{\frac{m}{2}}\frac{1}{2} & m \text{ is even} \end{cases} \right.$$
$$\left. -2 \begin{cases} \frac{m}{2}\left(2^{m-1} - \binom{m-1}{\frac{m-1}{2}}\right) & m \text{ is odd} \\ m2^{m-2} & m \text{ is even} \end{cases} \right)$$

$$= \left(\frac{1}{2}\right)^{m-1} B_2 \left(\begin{cases} m\binom{m-1}{\frac{m-1}{2}} & m \text{ is odd} \\ \frac{m}{2}\binom{m}{\frac{m}{2}} & m \text{ is even} \end{cases} \right)$$

By lemma 4 from Appendix 3.2 this becomes,

$$\leq \left(\frac{1}{2}\right)^{m-1} B_2 \left(\begin{cases} \frac{m2^{m-1}}{\sqrt{m}} & m \text{ is odd} \\ \frac{m2^{m-1}}{\sqrt{m+1}} & m \text{ is even} \end{cases} \right) \leq B_2\sqrt{m}$$

Substituting into 3.4 we get

$$\leq B_1\sqrt{m} \cdot \max_i \|\mathbf{x_i}\| + B_2\sqrt{m} \tag{3.6}$$

Dividing by m completes the proof

3.2 Combinatorial Lemmas

We will now prove the three lemmas used in the previous proof.

Lemma 4. *We'll now show that for m odd and $m \geq 3$*

$$\binom{m-1}{\frac{m-1}{2}} \leq \frac{2^{m-1}}{\sqrt{m}}$$

Proof: We will prove this by induction. Base case, $m = 3$

$$\binom{m-1}{\frac{m-1}{2}} = \binom{2}{1} \leq \frac{4}{\sqrt{3}}$$

Now, given

$$\binom{m-1}{\frac{m-1}{2}} \leq \frac{2^{m-1}}{\sqrt{m}} \tag{3.7}$$

We will show

$$\binom{m+1}{\frac{m+1}{2}} = \frac{(m+1)!}{\left(\left(\frac{m+1}{2}\right)!\right)^2} = \frac{(m+1)(m)(m-1)!}{\left(\left(\frac{m+1}{2}\right)\left(\frac{m-1}{2}!\right)\right)^2}$$

By 3.7

$$\leq \frac{(m+1)(m)}{\left(\frac{m+1}{2}\right)^2}\frac{2^{m-1}}{\sqrt{m}} = \frac{m}{(m+1)}\frac{2^{m+1}}{\sqrt{m}}$$

All that is left is to show

$$\frac{m}{(m+1)\sqrt{m}} \leq \frac{1}{\sqrt{m+2}}$$

Which is equivalent to showing

$$\left(\frac{m}{(m+1)}\right)^2 \leq \frac{m}{m+2}$$

So,

$$\left(\frac{m}{(m+1)}\right)^2 = \frac{m^2}{m^2+2m+1} \leq \frac{m^2}{m^2+2m} = \frac{m}{m+2}$$

Lemma 5. *For integer's* $m > 2$

$$\sum_{i=0}^{\lfloor \frac{m}{2} \rfloor} \binom{m}{i} = \begin{cases} 2^{m-1} & m \textit{ is odd} \\ 2^{m-1} + \binom{m}{\frac{m}{2}}\frac{1}{2} & m \textit{ is even} \end{cases} \tag{3.8}$$

Proof:

$$\sum_{i=0}^{\lfloor \frac{m}{2} \rfloor} \binom{m}{i} \tag{3.9}$$

When m is odd, by the symmetry of the binomial coefficients becomes

$$2^{m-1}$$

When m is even,

$$2^m = 2\sum_{i=0}^{\frac{m}{2}-1}\binom{m}{i} + \binom{m}{\frac{m}{2}}$$

Thus, 3.9 becomes

$$\sum_{i=0}^{\frac{m}{2}-1}\binom{m}{i} + \binom{m}{\frac{m}{2}} = \frac{1}{2}\left(2^m - \binom{m}{\frac{m}{2}}\right) + \binom{m}{\frac{m}{2}}$$

$$= 2^{m-1} + \binom{m}{\frac{m}{2}}\frac{1}{2}$$

Combining the even and odd parts for m, 3.9 becomes

$$\sum_{i=0}^{\lfloor\frac{m}{2}\rfloor}\binom{m}{i} = \begin{cases} 2^{m-1} & m \text{ is odd} \\ 2^{m-1} + \binom{m}{\frac{m}{2}}\frac{1}{2} & m \text{ is even} \end{cases} \tag{3.10}$$

Lemma 6. *For integer's* $m > 2$

$$\sum_{i=0}^{\lfloor\frac{m}{2}\rfloor}\binom{m}{i}i = \begin{cases} \frac{m}{2}\left(2^{m-1} - \binom{m-1}{\frac{m-1}{2}}\right) & m \textit{ is odd} \\ m2^{m-2} & m \textit{ is even} \end{cases} \tag{3.11}$$

Proof:

$$\sum_{i=0}^{\lfloor\frac{m}{2}\rfloor}\binom{m}{i}i = \sum_{i=1}^{\lfloor\frac{m}{2}\rfloor}\frac{m!}{i!(m-i)!}i = \sum_{i=1}^{\lfloor\frac{m}{2}\rfloor}\frac{m(m-1)!}{(i-1)!(m-i)!}$$

$$= m\sum_{i=1}^{\lfloor\frac{m}{2}\rfloor}\binom{m-1}{i-1} = m\sum_{i=0}^{\lfloor\frac{m}{2}\rfloor-1}\binom{m-1}{i} \tag{3.12}$$

As a reminder,

$$\sum_{i=0}^{m-1}\binom{m-1}{i} = 2^{m-1} \tag{3.13}$$

When m is even, $\lfloor \frac{m}{2} \rfloor = \frac{m}{2}$ and $\lfloor \frac{m-1}{2} \rfloor = \frac{m-1}{2} - \frac{1}{2} = \frac{m}{2} - 1$. By the symmetry of the binomial coefficients and the fact that $m-1$ is odd, half of 3.13 becomes

$$\sum_{i=0}^{\lfloor \frac{m-1}{2} \rfloor} \binom{m-1}{i} = \sum_{i=0}^{\frac{m}{2}-1} \binom{m-1}{i} = 2^{m-2}$$

Substituting into 3.12

$$m \sum_{i=0}^{\frac{m}{2}-1} \binom{m-1}{i} = m2^{m-2} \tag{3.14}$$

When m is odd, $\lfloor \frac{m}{2} \rfloor = \frac{m-1}{2}$ and $\lfloor \frac{m-1}{2} \rfloor = \frac{m-1}{2}$. By the symmetry of the binomial coefficient and the fact that $m-1$ is even, 3.13 becomes

$$\sum_{i=0}^{m-1} \binom{m-1}{i} = 2 \sum_{i=0}^{\frac{m-1}{2}-1} \binom{m-1}{i} + \binom{m-1}{\frac{m-1}{2}}$$

Thus,

$$\sum_{i=0}^{\frac{m-1}{2}-1} \binom{m-1}{i} = \frac{\left(2^{m-1} - \binom{m-1}{\frac{m-1}{2}}\right)}{2} \tag{3.15}$$

3.12 becomes

$$m \sum_{i=0}^{\frac{m-1}{2}-1} \binom{m-1}{i} = \frac{m}{2}\left(2^{m-1} - \binom{m-1}{\frac{m-1}{2}}\right)$$

Which completes the proof

3.3 Generalization of Hard-SVM with Bias

With our result on the Rademacher complexity of an affine class, we now present generalization bound for hard-SVM.

Theorem 7. *Suppose that $\mathcal{D}$ is a distribution over $\mathcal{X} \times \mathcal{Y}$ such that with probability 1 we have that $\|\mathbf{x}\| \leq R$. Let $\mathcal{H} = \{\mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle + b : \|\mathbf{w}\| \leq B_1, |b| \leq B_2\}$ and let $l : \mathcal{H} \circ Z \to \mathbb{R}$ be a loss function of the form $\phi(\langle \mathbf{w}, \mathbf{x} \rangle + b, y)$ such that $\forall y \in \mathcal{Y}$, $a \mapsto \phi(a, y)$ is a ρ-Lipschitz function and such that $max_{a \in [-B_1 R - B_2, B_1 R + B_2]} |\phi(a, y)| \leq c$. Then, for any $\gamma \in (0, 1)$, with probability of at*

least $1-\gamma$ *over the choice of an i.i.d. sample of size* m

$$\forall h \in \mathcal{H}, \quad \mathbb{E}_{\mathcal{D}}[\phi(h(\mathbf{x}), y)] \leq \mathbb{E}_{\mathcal{S}}[\phi(h(\mathbf{x}), y)] + \frac{2\rho(B_1 R + B_2)}{\sqrt{m}} + c\sqrt{\frac{2ln(2/\delta)}{m}} \tag{3.16}$$

Proof: Let $F = \{(\mathbf{x}, y) \mapsto \phi(h(\mathbf{x}), y) : h \in \mathcal{H}\}$. Then, $\mathcal{R}(F \circ \mathcal{S}) \leq \rho(\frac{B_1 R + B_2}{\sqrt{m}})$ by 3.3. And the theorem falls from [18] Thm. 26.5

Theorem 8. *Consider a distribution* $\mathcal{D}$ *over* $\mathcal{X} \times \{\pm 1\}$ *such that there exists some vector* $\mathbf{w}^\star$ *and some scalar* $b^\star$ *with* $\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[y\left(\langle \mathbf{w}^\star, \mathbf{x}\rangle + b^\star\right) \geq 1] = 1$ *and such that* $||\mathbf{x}||_2 \leq R$ *with probability 1. Let* $\mathbf{w}_\mathcal{S}, b_\mathcal{S}$ *be the output of hard-SVM. Then, with probability of at least* $1-\delta$ *over the choice of* $\mathcal{S} \sim \mathcal{D}^m$ *we have that*

$$\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[y \neq \text{sign}(\langle \mathbf{w}_\mathcal{S}, \mathbf{x}\rangle + b_\mathcal{S})] \leq \frac{8R||\mathbf{w}_\mathcal{S}|| + 2}{\sqrt{m}} + \sqrt{\frac{ln\left(\frac{4log_2(||\mathbf{w}_\mathcal{S}||)}{\delta}\right)}{m}} \tag{3.17}$$

Proof: Let $(\mathbf{x}_1)$ be one of the two closest points from the convex hull viewpoint of hard-SVM defined by Eqn. 4.3. Then,

$$b_\mathcal{S} = 1 - \langle \mathbf{w}_\mathcal{S}, \mathbf{x_1}\rangle$$

And since,

$$|\langle \mathbf{w}_\mathcal{S}, \mathbf{x_1}\rangle| \leq \|\mathbf{w}_\mathcal{S}\| \, \|\mathbf{x_1}\| \leq \|\mathbf{w}_\mathcal{S}\| \, R$$

This implies,

$$|b_\mathcal{S}| \leq 1 + \|\mathbf{w}_\mathcal{S}\| \, R$$

For any integer i, let $\beta_i = 2^i$, $\mathcal{H}_i = \{\mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x}\rangle + b : \|\mathbf{w}\| \leq \beta_i, |b| \leq 1 + \beta_i R\}$ and let

$\delta_i = \frac{\delta}{2i^2}$. Fix i, then by 7, we have with probability of at least $1 - \delta_i$

$$\forall h \in \mathcal{H}, \quad \mathbb{E}_{\mathcal{D}}[\phi(h(\mathbf{x}), y)] \leq \mathbb{E}_{\mathcal{S}}[\phi(h(\mathbf{x}), y)] + \frac{2\rho(\beta_i R + \beta_i R)}{\sqrt{m}} + c\sqrt{\frac{2\ln(2/\delta)}{m}} \tag{3.18}$$

By using the ramp loss function that has a Lipschitz constant $\rho = 1$

$$\phi(x) = \begin{cases} 1 & x \leq 0 \\ 1 - x & 0 < x \leq 1 \\ 0 & x > 1 \end{cases}$$

We get

$$\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[y \neq \text{sign}(\langle \mathbf{w}_{\mathcal{S}}, \mathbf{x} \rangle + b_{\mathcal{S}})] \leq \mathbb{E}_{\mathcal{D}}[\phi(h(\mathbf{x}), y)]$$

And

$$\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{S}}[y(\langle \mathbf{w}_{\mathcal{S}}, \mathbf{x} \rangle + b_{\mathcal{S}}) \geq 1] \geq \mathbb{E}_{\mathcal{S}}[\phi(h(\mathbf{x}), y)]$$

Using the fact that $\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{S}}[y(\langle \mathbf{w}_{\mathcal{S}}, \mathbf{x} \rangle + b_{\mathcal{S}}) \geq 1] = 0$ for the output of hard-SVM and applying the union bound and using $\sum_{i=1}^{\infty} \delta_i \leq \delta$ we obtain with probability of at least $1 - \delta$ this holds for all i. Therefore, for all h, if we let $i = \lceil \log_2(\|\mathbf{w}\|) \rceil$ then $h \in \mathcal{H}_i$, $\beta_i \leq 2\|\mathbf{w}\|$ and $\frac{2}{\delta_i} = \frac{(2i)^2}{\delta} \leq \frac{(4\log_2(\|\mathbf{w}\|))^2}{\delta}$. Therefore,

$$\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[y \neq \text{sign}(\langle \mathbf{w}_{\mathcal{S}}, \mathbf{x} \rangle + b_{\mathcal{S}})] \leq \frac{8R||\mathbf{w}_{\mathcal{S}}|| + 2}{\sqrt{m}} + \sqrt{\frac{\ln\left(\frac{4\log_2(||\mathbf{w}_{\mathcal{S}}||)}{\delta}\right)}{m}} \tag{3.19}$$

4. COMPRESSED LEARNING WITH HARD-SVM WHEN SPARSE

4.1 Introduction

Compressed learning is the process of running machine learning algorithms on data that has been compressed. There are many potential reasons to run machine learning algorithms directly on the compressed data. These reasons include: 1) to reduce the computation time of the algorithm; 2) to reduce the storage cost of the algorithm; 3) to reduce the data transmission cost of server/node configurations.

Running machine learning algorithms on compressed data to reduce the computation time and storage costs of the algorithms can be motivated by the size of datasets seen today. With the popularity of machine learning sky-rocketing these days, machine learning is being applied to more and more scenarios. As such, the size of the datasets is exploding as well. For example, the ODP dataset with 422,712 features for each data point in each of the 105,033 classes, would require more than 160GB to store a standard logistic regression model [7]. Reducing the number of features through compression has the potential to greatly reduce this model size while still using a standard logistic regression model. Furthermore, while a hashing scheme such as Weinberger et al. [21] can mitigate the run-time dependence on feature size, most algorithms do not have this capability and would run significantly faster with a reduced feature size.

The goal of reducing the data transmission cost in server/node configurations can be seen in many applications such as traffic analysis based on packet timing and packet sizes. With encryption becoming more and more common, content-based traffic analysis is no longer possible. Researchers have been exploring the use of traffic analysis based on packet timing and sizes to defend or attack networks. In the basic setup for this line of research, many network relay nodes send packet statistics to a central server. The central server uses machine learning techniques to identify the anonymized users and attacks. Ideally, these are identified in real-time. The data that the nodes send to the central server must be compressed based on the channel capacity. Although

some compression techniques allow for the recovery of the original signal, the real-time constraint of the analysis makes it impractical for the server to reconstruct each signal forcing the server to run the machine learning algorithm on the compressed data. [16]

Although there are many signal compression techniques, we will restrict our attention in this paper to the techniques from compressive sensing, which are backed by a rich theory. For compressed sensing, there is a compression matrix $\Phi \in \mathbb{R}^{l \times n}$ and an original signal $x \in \mathbb{R}^n$. The observed signal is $z = \Phi x$. Though classical linear algebra as well as Shannon's sampling theorem imply $l \geq n$ for the reconstruction of x, it has been shown in compressed sensing that, if x is s-sparse, $l \geq Cs \ln(n/s)$ is sufficient to recover x for some constant c. Thus sparse signals can be linearly compressed by a log factor [11].

Though compression is necessary and beneficial in many scenarios, we must be assured that compression will not impair the goal of using machine learning in the first place – to obtain an output function from a function class that achieves a low probability of prediction error with respect to the unknown underlying distribution. One popular function class from which to obtain an output function is the class of linear classifiers. It is easy to imagine a scenario where a linear classifier performs poorly after compression. An extreme example of this is the compression of signals by Φ to zero vectors with length 1.

One of the most popular and theoretically sound algorithms for choosing an output function from a linear function class is support vector machines (SVM). The basic idea of SVM is the concept of margins. For a given data point and a linear classifier, the margin is the signed distance from the point to the hyperplane represented by the classifier. The margin is positive if the point is classified correctly and negative otherwise. Two variants of SVM have been studied: hard-SVM and soft-SVM. Hard-SVM maximizes the minimum margin of all points in the training set guaranteeing all points are classified correctly. This requires the assumption that it is possible to classify all points correctly with a linear classifier (linear separability). Soft-SVM allows for some points to be incorrectly classified but penalizes the margins of the violations. Soft-SVM does not require linear separability [18].

4.1.1 Previous Work

Calderbank, Jafarpour, and Schapire [4] extensively analyzed how compressed learning affects the generalization of soft-SVM. They showed that if the domain space is sparse and the compression matrix satisfies certain conditions, the hinge loss of soft-SVM on the compressed data is close (in a precise sense) to the hinge loss of soft-SVM on the uncompressed data.

4.1.2 Our Contributions

In this chapter, we give theoretical bounds on the generalization of compressed hard-SVM in terms of the restricted isometric constant of the compression matrix Φ. This allows users to choose how much to compress the data and still be provably within the design criteria at hand. This analysis answers the same question for hard-SVM that Calderbank et al. [4] answered for soft-SVM, but through a significantly different proof technique based on the geometry of hard-SVM . Furthermore, the analysis of hard-SVM requires additional steps when compared with Calderbank et al. [4] because hard-SVM requires the linear separability assumption to be valid while soft-SVM can work with any dataset. This means that we have to establish criteria on allowable compression matrices before analyzing how the generalization bound changes with respect to the compression matrix used.

4.1.3 Organization

The rest of the chapter is organized as follows. Next in section 4.2, we formally state the problem and review some basic results on hard-SVM and compressive sensing. The main results of the paper are presented in section 4.3. In particular, Theorem 9 establishes a criterion on the compression matrix to preserve linear separability. Theorem 10 provides a generalization bound for compressed learning with hard-SVM. We discuss the issues of sensitivity and prior knowledge for these results in section 4.4. The following three sections, 4.5, 4.6, and 4.7, provide the proofs of the two main results. Section 4.5 presents two lemma's relating to the compression matrix necessary in the proofs. Section 4.6 shows that if there exists a linearly separable solution, there also exists a linearly separable solution that behaves nicely with compression. This is the most technical

intermediary result necessary for the proof of the main results. Finally, section 4.7 provides the proofs of the main results.

4.2 Problem Setup

Let the domain, $\mathcal{X}$, of the classification task be a Euclidean space of dimension d. That is, let $\mathcal{X} = \mathbb{R}^d$. Let the label space, $\mathcal{Y}$, of the classification task be $\{+1, -1\}$, as binary classification is the assumption by SVM. Note, that multiclass classification can be reduced to this binary setup [9, 1]. Let $\mathcal{M} = \mathcal{B}_{\mathbb{R}^d} \otimes \mathcal{P}(\{+1, -1\})$ be the σ-algebra on which the probability measure is defined where $\mathcal{B}_{\mathbb{R}^d}$ is the borel σ-algebra on $\mathbb{R}^d$ and $\mathcal{P}$ denotes the power sets. Altogether, let $(\mathcal{X} \times \mathcal{Y}, \mathcal{M}, \mu)$ be a probability measure space.

Let the training set (sample data) $\mathcal{S} = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m))$ be drawn according to $\mu^{\otimes m}$. The hard-SVM procedure is defined via solving the following optimization problem [18]

$$(\mathbf{w}_{\mathcal{S}}, b) = \underset{(\mathbf{w},b)}{\operatorname{argmin}} ||\mathbf{w}||^2 \tag{4.1}$$

$$\text{s.t.} \quad \forall i, \quad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 \tag{4.2}$$

Note that for above optimization problem to be feasible for all training sets, the underlying distribution μ must be linearly separable. More precisely, there must exist $\mathbf{w}^\star$ and $b^\star$ such that $\mathbb{P}_{(\mathbf{x},y)\sim\mu}[y\,(\langle \mathbf{w}^\star, \mathbf{x} \rangle + b^\star) \geq 1] = 1$.

4.2.1 Convex Hull Formulation

This paper will rely on the ideas developed by Bennett and Bredensteiner [3], which showed solving hard-SVM is equivalent to finding the two closest points in the two convex hulls of the training set. We will explain this precisely next.

Let $\mathcal{S}^1 := \{(\mathbf{x}, y) \in \mathcal{S} : y = 1\}$ and $\mathcal{S}^{-1} := \{(\mathbf{x}, y) \in \mathcal{S} : y = -1\}$ be the set of training examples where y is 1 and -1 respectively. Additionally, $\mathcal{S}_x^1 := \{\mathbf{x} : (\mathbf{x}, y) \in \mathcal{S}^1\}$ and $\mathcal{S}_x^{-1} := \{\mathbf{x} : (\mathbf{x}, y) \in \mathcal{S}^{-1}\}$ be the set of x values in each training set partition. For an arbitrary set $\mathcal{A}$, let $\text{co}(\mathcal{A})$

denote the convex hull of $\mathcal{A}$. That is, $\text{co}(\mathcal{A}) := \left\{\sum_{i=1}^{n} \alpha_i \mathbf{x}_i : n \leq |\mathcal{A}|,\ \mathbf{x}_i \in \mathcal{A},\ \alpha \in \mathbb{R}_+^n,\ \|\alpha\|_1 = 1\right\}$

Then, the two closest points on the convex hull's becomes

$$(\mathbf{x}_1, \mathbf{x}_2) = \underset{\mathbf{x}_1 \in \text{co}(\mathcal{S}^1),\ \mathbf{x}_2 \in \text{co}(\mathcal{S}^{-1})}{\text{argmin}} \|\mathbf{x}_1 - \mathbf{x}_2\|^2 \tag{4.3}$$

and the corresponding solution to hard-SVM is given by:

$$\mathbf{w}_{\mathcal{S}} = \frac{2(\mathbf{x}_1 - \mathbf{x}_2)}{\|\mathbf{x}_1 - \mathbf{x}_2\|^2} \tag{4.4}$$

and

$$b = 1 - \frac{2\langle \mathbf{x}_1 - \mathbf{x}_2, \mathbf{x}_1 \rangle}{\|\mathbf{x}_1 - \mathbf{x}_2\|^2} \tag{4.5}$$

4.2.2 Compression

The following result is known in compressive sensing. If a signal x is s-sparse, a sensing matrix $\Phi \in \mathbb{R}^{l \times n}$ can be found such that $z = \Phi x$ is a one-to-one map for all s-sparse x when $l \geq Cs \ln(n/s)$. We present the following definition from [11].

Definition 1. The s-th *restricted isometry constant* $\delta_s = \delta_s(\Phi)$ of a matrix $\Phi \in \mathbb{R}^{l \times n}$ is the smallest $\delta > 0$ such that

$$(1 - \delta)\|\mathbf{x}\|^2 \leq \|\Phi \mathbf{x}\|^2 \leq (1 + \delta)\|\mathbf{x}\|^2 \tag{4.6}$$

for all s-sparse $\mathbf{x} \in \mathbb{R}^n$. Equivalently, it is given by

$$\delta_s = \max_{S \subset [N], |S| \leq s} \|\Phi_S^\intercal \Phi_S - I_{n \times n}\|_{2 \to 2} \tag{4.7}$$

where Φ_S is Φ restricted to the rows of S and $\|.\|_{2 \to 2}$ is the operator norm.

4.3 Main Results

4.3.1 Compression Limits

In this section, we present the first of our two main results. We give bounds on the compression ratios of the compression matrices that preserve linear separability. We will do this by bounding the restricted isometric constant of the compression matrix Φ. This in turn will bound the compression ratio. This step, the maximum allowable compression, is not something Calderbank et al. [4] had to analyze for soft-SVM as soft-SVM can work in any scenario. Hard-SVM requires the linear separability assumption which requires this analysis. The main result is stated below and mostly follows from the details in section 4.6 on the existence of a sparse-like solution.

Theorem 9. *If μ satisfies $\mathbb{P}_{(\mathbf{x},y)\sim\mu}[\|x\|_0 \leq s] = 1$ and $\mathbb{P}_{(\mathbf{x},y)\sim\mu}[\|x\|_2 \leq R] = 1$ and if $\exists \mathbf{w}^\star, b^\star$ such that $\mathbb{P}_{(\mathbf{x},y)\sim\mu}[y\left(\langle \mathbf{w}^\star, \mathbf{x}\rangle + b^\star\right) \geq 1] = 1$, a compression matrix Φ with a $2s$-restricted isometric constant less than $\frac{1}{\|\mathbf{w}^\star\|^2 R^2}$ is linearly separable in the compressed domain. That is, $\exists \mathbf{w}_C \in \mathbb{R}^l, b_C \in \mathbb{R}$ such that*

$$\mathbb{P}_{(\mathbf{x},y)\sim\mu}[y(\langle \mathbf{w}_C, \Phi\mathbf{x}\rangle + b_C) \geq 1] = 1$$

Theorem 9 begins with conditions on the probability measure μ. As we need the sparsity of the domain vectors for the theory of compressive sensing to be satisfied, the first condition on μ states that x must be s-sparse with probability 1. The next two conditions on μ are standard conditions for the analysis of hard-SVM. That is, x must be bounded with probability 1 and there must exists a linearly separable $\mathbf{w}^\star, b^\star$ that predicts y with probability 1.

The theorem states that if these conditions are satisfied and the isometric constant of the compression matrix is satisfactory, then the compressed domain is also linearly separable. This means the conditions necessary for hard-SVM are satisfied in the compressed domain and we can run the algorithm expecting the correct performance.

We now briefly discuss how to translate this result to the compressed vector length (compression ratio). In Thm. 6.8 of Foucart and Rauhut [11], it is shown than the compression length

of matrix must be greater than $c\frac{s}{\delta_{2s}^2}$ for a constant c detailed in the book. Furthermore, it is shown in Thm. 9.9 of Foucart and Rauhut [11] that a subgaussian matrix with compression length $\geq \frac{C}{\delta_{2s}}(7s + 2\ln(2\epsilon^{-1}))$ will have the necessary restricted isometric constant with high probability. Overall, a increase in the allowed δ_{2s} results in a squared decrease of the compressed vector length.

4.3.2 Compressed Generalization Bounds

We now present the second of our main results for this paper. We show how the generalization bound changes with respect to the $2s$-restricted isometric constant of the compression matrix Φ. For ease of notation, we denote for a fixed m, R, $\mathbf{w}_{\mathcal{S}}$, δ, and s the generalization bound (the bound on the probability of error) as $\mathcal{L}_B$. We derive this generalization bound in the appendix by extending the previously published results for hard-SVM of linear classes to one with affine classes [13, 18]. Precisely, $\mathcal{L}_B$ is defined

$$\mathcal{L}_B = \frac{8R||\mathbf{w}_{\mathcal{S}}|| + 2}{\sqrt{m}} + \sqrt{\frac{\ln\left(\frac{4\log_2(||\mathbf{w}_{\mathcal{S}}||)}{\delta}\right)}{m}}$$

We will show how the probability of error in the compressed domain scales with respect to $\mathcal{L}_B$ in terms of the restricted isometric constant of Φ.

Theorem 10. *If μ and Φ satisfy the conditions of Thm. 9 and $\mathbf{w}_{\mathcal{S}}^{\Phi}, b_{\mathcal{S}}^{\Phi}$ is the output of hard-SVM on the compressed data, then with probability $\geq 1 - \delta$*

$$\mathbb{P}_{(\mathbf{x},y)\sim\mu}[y \neq \text{sign}(\langle \mathbf{w}_{\mathcal{S}}^{\Phi}, \Phi\mathbf{x}\rangle + b_{\mathcal{S}}^{\Phi})] \leq \frac{\mathcal{L}_B}{\sqrt{1 - \delta_{2s}R^2\left\|\mathbf{w}_{\mathcal{S}}\right\|^2}}$$

It is important to note that the multiplicative scaling term in the theorem is always greater or equal to 1. That is, compression cannot improve the generalization bound which is expected from an intuitive standpoint.

4.4 Discussion

4.4.1 Sensitivity

We will briefly compare the compressed generalization bounds of hard-SVM to the soft-SVM. In Calderbank et al. [4], they show the hinge loss $H_{\mathcal{D}}(.)$ of the compressed learned soft-SVM hyperplane $\mathbf{w}_{\mathcal{S}}^{\Phi}$ is close to the hinge loss of the uncompressed soft-SVM hyperplane $\mathbf{w}_{\mathcal{S}}$. Precisely, with probability $1 - 2\delta$

$$H_{\mathcal{D}}(\mathbf{w}_{\mathcal{S}}^{\Phi}) \leq H_{\mathcal{D}}(\mathbf{w}_{\mathcal{S}}) + O\left(\sqrt{\|\mathbf{w}_{\mathcal{S}}\|^2 \left(R^2\delta_{2s} + \frac{\log(1/\delta)}{m}\right)}\right)$$

The sensitivity to an increase in δ_{2s} is the derivative of the generalization bound with respect to δ_{2s} at δ_{2s}. The sensitivity is the penalty paid for an arbitrary increase in the restricted isometric constant. For soft-SVM, the sensitivity S_{soft} is

$$S_{\text{soft}} = \frac{\|\mathbf{w}_{\mathcal{S}}\|^2 R^2}{2\sqrt{\|\mathbf{w}_{\mathcal{S}}\|^2 \left(R^2\delta_{2s} + \frac{\log(1/\delta)}{m}\right)}}$$

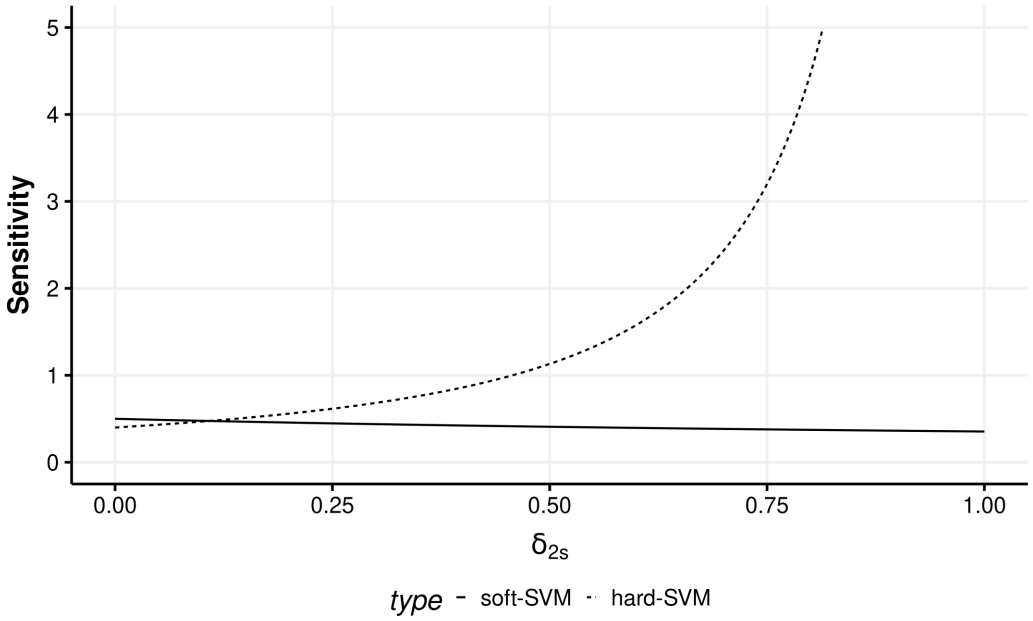
Whereas the sensitivity for hard-SVM is

$$S_{\text{hard}} = \frac{\mathcal{L}_B R^2 \|\mathbf{w}_{\mathcal{S}}\|^2}{2\left(1 - \delta_{2s}R^2 \|\mathbf{w}_{\mathcal{S}}\|^2\right)^{3/2}}$$

Before contrasting the sensitivities of compression, we will note the intuitive differences in the two algorithms. From an intuitive standpoint, hard-SVM is more sensitive to a perturbation of a single training point than soft-SVM. Soft-SVM can simply incur a slight cost for the perturbation while hard-SVM is forced to adapt to the perturbation.

This intuition is mirrored in the actual sensitivities of hard and soft SVM. Hard-SVM has a sensitivity that grows significantly. It is interesting to note that the original generalization bound does not show up in the sensitivity of soft-SVM while is present in the sensitivity of hard-SVM.

Figure 4.1: Sensitivity of Hard/Soft-SVM



Thus, depending on the current generalization bound, the sensitivity of hard-SVM can actually be initially lower than the sensitivity of soft-SVM. In Fig. 4.1, we plot the two sensitivities with $R = 1$, $\|\mathbf{w}_{\mathcal{S}}\| = 1$, $\mathcal{L}_B = 1$, and $\frac{1/\delta}{M} = 1$ to highlight the differing growth rates with respect to δ_{2s}.

4.4.2 Prior Knowledge

The reliance on the unknown $\mathbf{w}^\star$ is also a unique aspect of compressed learning for hard-SVM. While both Thm. 10 and 9 are stated with respect to $\mathbf{w}^\star$, we show in the proof of Thm. 10 that it can be written in terms of the known $\mathbf{w}_{\mathcal{S}}$. We note this is the same as in the analysis of soft-SVM. That is, the performance drop-off due to compression can be calculated in both hard-SVM and soft-SVM through the use of $\mathbf{w}_{\mathcal{S}}$. However, the results of the analysis of allowable compression cannot be stated in terms of $\mathbf{w}_{\mathcal{S}}$ and rely solely on $\mathbf{w}^\star$. We note that soft-SVM did not require any such analysis.

This brings us to the crux of hard-SVM: hard-SVM requires the separability assumption. The assumption is prior knowledge and cannot be estimated by the data set at hand. It must be assumed.

Thus, it is not entirely surprising that the compression range depends on a variable that must be assumed beforehand as well. While the exact $\mathbf{w}^\star$ must not be assumed, some upper bound on its norm must be.

Ideally, it would be nice if we could state a theorem that says with probability $1-\delta$, the difference between the norms of $\mathbf{w}_\mathcal{S}$ and $\mathbf{w}^\star$ is smaller than $f(m,\delta)$ where $f(m,\delta)$ is a decreasing function on m and increasing function in δ. But this cannot be done. We can show the impossibility of this result with a simple example.

Let there be 4 points in the distribution's support with positive measure: $(\mathbf{x}_1, y_1) = ([R, 0, 0, \ldots], 1)$, $(\mathbf{x}_2, y_2) = ([\gamma, 0, 0, \ldots], 1)$, $(\mathbf{x}_3, y_3) = ([-R, 0, 0, \ldots], -1)$, $(\mathbf{x}_4, y_4) = ([-\gamma, 0, 0, \ldots], -1)$ for some $\gamma \in (0, R)$. Let $\mathbb{P}[x_1] = (1-\phi)/2$, $\mathbb{P}[x_3] = (1-\phi)/2$, $\mathbb{P}[x_2] = \phi/2$, $\mathbb{P}[x_4] = \phi/2$ for some $\phi \in (0,1)$. Note that these data points satisfy the sparsity, bounded norm and linear separability assumptions required in our main results.

The norm of $\mathbf{w}^\star$ is $1/\gamma$. However, if x_2 and x_4 are not both included in the sample set, the norm of $\mathbf{w}_\mathcal{S}$ is less than $2/R$. If both x_2 and x_4 are in the sample set, the norm of $\mathbf{w}_\mathcal{S}$ is $1/\gamma$. The probability of the sample set of size m containing both x_2 and x_4 is less than the probability that it contains either x_2 and x_4 which is $1-(1-\phi)^m$. Note that the infimum over $\phi \in (0,1)$ of this probability is 0. As this is for any arbitrary m we have

$$\lim_{n\to\infty} \sup_{\phi\in(0,1)} \mathbb{P}_\mu \left[|\|\mathbf{w}_\mathcal{S}\| - \|\mathbf{w}^\star\|| \geq 1/\gamma - 2/R \right] = 1$$

Noting that $\sup_{\gamma\in(0,R)} 1/\gamma = \infty$ implies that we cannot bound the difference between the norms $\mathbf{w}_\mathcal{S}$ and $\mathbf{w}^\star$ by any finite amount with any probability.

4.5 Compressed Vectors Lemmas

We will now give two lemmas relating to compression that are required for the complete proof of the main results from this paper. The first lemma characterizes how the inner product of compressed s-sparse vectors relates to their original inner product.

Lemma 11. *Let $\mathbf{x_1}, \mathbf{x_2} \in \{\mathbf{x} \in \mathbb{R}^d : ||\mathbf{x}||_0 \leq s\}$ be two s-sparse vectors of dimension d. Then,*

$$|\langle \Phi \mathbf{x_1}, \Phi \mathbf{x_2} \rangle - \langle \mathbf{x_1}, \mathbf{x_2} \rangle| \leq \delta_{2s} \left\| \mathbf{x_1} \right\|_2 \left\| \mathbf{x_2} \right\|_2 \tag{4.8}$$

Proof. Let $S = \text{supp}(\mathbf{x_1}) \cup \text{supp}(\mathbf{x_2})$ where $\text{supp}(\mathbf{x})$ is the support of $\mathbf{x}$ (indices on which $\mathbf{x}$ is nonzero). Then, $|S| \leq 2s$ and

$$\begin{aligned}
& |\langle \Phi \mathbf{x_1}, \Phi \mathbf{x_2} \rangle - \langle \mathbf{x_1}, \mathbf{x_2} \rangle| \\
= \; & |\langle \Phi_S \mathbf{x_1}_S, \Phi_S \mathbf{x_2}_S \rangle - \langle \mathbf{x_1}_S, \mathbf{x_2}_S \rangle| \\
= \; & |\langle (\Phi_S^\intercal \Phi_S - I_{2s \times 2s}) \mathbf{x_1}_S, \mathbf{x_2}_S \rangle| \\
\leq \; & \left\| (\Phi_S^\intercal \Phi_S - I_{2s \times 2s}) \mathbf{x_1}_S \right\| \left\| \mathbf{x_2}_S \right\| \\
\leq \; & \left\| (\Phi_S^\intercal \Phi_S - I_{2s \times 2s}) \right\|_{2 \to 2} \left\| \mathbf{x_1}_S \right\| \left\| \mathbf{x_2}_S \right\| \\
\leq \; & \delta_{2s} \left\| \mathbf{x_1}_S \right\| \left\| \mathbf{x_2}_S \right\| \\
\leq \; & \delta_{2s} \left\| \mathbf{x_1} \right\| \left\| \mathbf{x_2} \right\|
\end{aligned}$$

The second lemma characterizes how distances between points that are *convex combinations* of sparse vectors relate before and after compression.

Lemma 12. *Let $\mathbf{x}_1$ be a convex combination of s-sparse vectors. That is, let $\mathbf{x}_1 = \sum_{i=1}^{n_1} \alpha_i \tilde{\mathbf{x}}_i$ where $n_1 \in \mathbb{N}$, $\alpha \in \mathbb{R}_+^{n_1}$ and $\|\alpha\|_1 = 1$, and $\|\tilde{\mathbf{x}}_i\|_0 \leq s$ and $\|\tilde{\mathbf{x}}_i\| \leq R$ for all $i \in [n_1]$. Similarly, let $\mathbf{x}_2$ be a convex combination of s-sparse vectors. That is, let $\mathbf{x}_2 = \sum_{i=1}^{n_2} \beta_i \ddot{\mathbf{x}}_i$ where $n_2 \in \mathbb{N}$, $\beta \in \mathbb{R}_+^{n_2}$ and $\|\beta\| = 1$, and $\|\ddot{\mathbf{x}}_i\|_0 \leq s$ and $\|\ddot{\mathbf{x}}_i\| \leq R$ for all $i \in [n_2]$. Then,*

$$\left\| \Phi \mathbf{x_1} - \Phi \mathbf{x_2} \right\|^2 \geq \left\| \mathbf{x_1} - \mathbf{x_2} \right\|^2 - 4R^2 \delta_{2s} \tag{4.9}$$

Proof.

$$\|\Phi\mathbf{x_1} - \Phi\mathbf{x_2}\|^2 = \left\|\sum_{i=1}^{n_1}\alpha_i\Phi\tilde{\mathbf{x}}_i - \sum_{i=1}^{n_2}\beta_i\Phi\ddot{\mathbf{x}}_i\right\|^2$$

$$= \left\|\sum_{i=1}^{n_1}\alpha_i\Phi\tilde{\mathbf{x}}_i\right\|^2 + \left\|\sum_{i=1}^{n_2}\beta_i\Phi\ddot{\mathbf{x}}_i\right\|^2 - 2\left\langle\sum_{i=1}^{n_1}\alpha_i\Phi\tilde{\mathbf{x}}_i, \sum_{i=1}^{n_2}\beta_i\Phi\ddot{\mathbf{x}}_i\right\rangle \quad (4.10)$$

The first term on the right-hand side of 4.10 becomes (a) by lemma 11 for sparse vectors, (b) by the norm bound on $\mathbf{x}$, (c) by rearranging

$$\begin{aligned}
\left\|\sum_{i=1}^{n_1}\alpha_i\Phi\tilde{\mathbf{x}}_i\right\|^2 &= \sum_{i=1}^{n_1}\sum_{j=1}^{n_1}\alpha_i\alpha_j\left\langle\Phi\tilde{\mathbf{x}}_i, \Phi\tilde{\mathbf{x}}_j\right\rangle \\
&\geq \sum_{i=1}^{n_1}\sum_{j=1}^{n_1}\alpha_i\alpha_j\left(\left\langle\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j\right\rangle - \delta_{2s}\left\|\tilde{\mathbf{x}}_i\right\|\left\|\tilde{\mathbf{x}}_j\right\|\right) && \text{(a)} \\
&\geq \sum_{i=1}^{n_1}\sum_{j=1}^{n_1}\alpha_i\alpha_j\left\langle\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j\right\rangle - \delta_{2s}R^2 && \text{(b)} \\
&= \left\|\sum_{i=1}^{n_1}\alpha_i\tilde{\mathbf{x}}_i\right\|^2 - \delta_{2s}R^2 && \text{(c)}
\end{aligned}$$

Similarly for the second term

$$\left\|\sum_{i=1}^{n_2}\beta_i\Phi\ddot{\mathbf{x}}_i\right\|^2 \geq \left\|\sum_{i=1}^{n_2}\beta_i\ddot{\mathbf{x}}_i\right\|^2 - \delta_{2s}R^2$$

Now, the third term on the right-hand side of 4.10 becomes (a) by the linearity of inner products,

(b) by lemma 11 for sparse vectors, (c) by the norm bound on $\mathbf{x}$ and the linearity of inner products

$$\left\langle \sum_{i=1}^{n_1} \alpha_i \Phi \tilde{\mathbf{x}}_i, \sum_{i=1}^{n_2} \beta_i \Phi \ddot{\mathbf{x}}_i \right\rangle = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \alpha_i \beta_j \left\langle \Phi \tilde{\mathbf{x}}_i, \Phi \ddot{\mathbf{x}}_j \right\rangle \tag{a}$$

$$\leq \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \alpha_i \beta_j \left(\langle \tilde{\mathbf{x}}_i, \ddot{\mathbf{x}}_j \rangle + \delta_{2s} \left\| \tilde{\mathbf{x}}_i \right\| \left\| \ddot{\mathbf{x}}_\mathbf{j} \right\| \right) \tag{b}$$

$$\leq \left\langle \sum_{i=1}^{n_1} \alpha_i \tilde{\mathbf{x}}_i, \sum_{i=1}^{n_2} \beta_i \ddot{\mathbf{x}}_i \right\rangle - \delta_{2s} R^2 \tag{c}$$

Combining the bounds for each of the three terms in 4.10 and pulling out the definition of $\left\| \mathbf{x}_1 - \mathbf{x}_2 \right\|^2$ we complete the proof.

4.6 Existence of Sparse-Like Solution

Finally, we present the intermediate result required for the compression limits bound. Precisely, we show that if there exists a separable $\mathbf{w}^\star, b^\star$ there exists another separable pair $\mathbf{w}_0, b_0$ that behave nicely with compression. This result is needed for the analysis of the allowable compression range.

The compressed sensing theory shows us that norms, inner products and distances are approximately preserved when compressing with appropriate matrices. However, as we have no sparsity assumptions on $\mathbf{w}^\star$ we cannot apply those theorems and there will be no guarantees for $\langle \Phi \mathbf{w}^\star, \Phi \mathbf{x} \rangle$ in terms of $\langle \mathbf{w}^\star, \mathbf{x} \rangle$ that we would have if $\mathbf{w}^\star$ were sparse. To get around this, we will show that if there exists linearly separable $\mathbf{w}^\star$ and $b^\star$ then there exists linearly separable $\mathbf{w}_0, b_0$ that satisfy an inner product bound in terms of the restricted isometric constant. This is a fairly technical result as the underlying support of the distribution μ may be abstract.

Proposition 13. *If μ satisfies $\mathbb{P}_{(\mathbf{x},y)\sim\mu}[\left\| x \right\|_0 \leq s] = 1$ and $\mathbb{P}_{(\mathbf{x},y)\sim\mu}[\left\| x \right\|_2 \leq R] = 1$ and if $\exists \mathbf{w}^\star, b^\star$ such that $\mathbb{P}_{(\mathbf{x},y)\sim\mu}[y \left(\langle \mathbf{w}^\star, \mathbf{x} \rangle + b^\star \right) \geq 1] = 1$. Then, there exist $\mathbf{w}_0$ such that $\mathbb{P}_{(\mathbf{x},y)\sim\mu}[y \left(\langle \mathbf{w_0}, \mathbf{x} \rangle + b^\star \right) \geq 1] = 1$, $\left\| \mathbf{w}_0 \right\| \leq \left\| \mathbf{w}^\star \right\|$ and for all s-sparse $\mathbf{x} \in \mathbb{R}^d$*

$$\left| \langle \Phi \mathbf{w}_0, \Phi \mathbf{x} \rangle - \langle \mathbf{w}_0, \mathbf{x} \rangle \right| \leq \delta_{2s} \left\| \mathbf{w}_0 \right\|^2 R^2 \tag{4.11}$$

This proposition will require several steps to prove so first we will go through some notation to be used throughout.

4.6.0.0.1 Notation Let $\mathcal{G}^+$ be the support of μ when $y = 1$. That is,

$$\mathcal{G}^+ = \left\{x : \forall \mathcal{N} \subseteq \mathbb{R}^d \text{ open } \& \; x \in \mathcal{N} \Rightarrow \mu((\mathcal{N}, 1)) > 0\right\}$$

Similarly, let $\mathcal{G}^-$ be the support of μ when $y = -1$.

$$\mathcal{G}^- = \left\{x : \forall \mathcal{N} \subseteq \mathbb{R}^d \text{ open } \& \; x \in \mathcal{N} \Rightarrow \mu((\mathcal{N}, -1)) > 0\right\}$$

We will assume that we are not in a degenerate case where either set is the empty set. Note that this implies $\mathbf{w}^\star$ cannot be the all-zero vector.

Additionally, for a set $\mathbf{A}$ we will denote cl($\mathcal{A}$) to be the closure of $\mathcal{A}$.

The basic structure of the proof is the following

1. We show if we have linear separability, we know geometrically that the distance between any two points in convex hulls of the two supports is bounded away from zero
2. We show that the support of the distribution we have defined has measure 1
3. We show that for each point in the support, the sparsity, bounded norm and linear separability assumptions hold
4. We show that two points in the closure of the convex hulls of the supports achieves the infimum of the distance between the two convex hulls
5. We show that if the distance between any two points in convex hulls of the two supports is bounded away from zero, we can construct a solution based on the two convex hulls
6. We show the solution we constructed this way satisfies Eqn. 4.11.

First, we show that if we have linear separability, we get that the distance between any two points in convex hulls of the two supports is bounded away from zero

Step 1. *If the conditions of Prop. 13 are satisfied. Then,* $\exists \delta > 0$ *such that*

$$\inf \left\{ \|\mathbf{x}_1 - \mathbf{x}_2\| : \mathbf{x}_1 \in \text{co}(\mathcal{G}^+), \mathbf{x}_2 \in \text{co}(\mathcal{G}^-) \right\} = \delta$$

Proof. To prove this theorem we first need to show the following proposition that relates the conditions of Prop. 13 to the supports defined previously.

Step 2. *Let* $\mathcal{G}_y = (\mathcal{G}^+ \times \{1\}) \cup (\mathcal{G}^- \times \{-1\})$

$$\mu(\mathcal{G}_y) = 1 \tag{4.12}$$

$$\mu\left(\mathbb{R}^d \times \{1, -1\} \setminus \mathcal{G}_y\right) = 0 \tag{4.13}$$

4.13 is true by the fact $\mathbb{R}^d$ is a Hausdorff space, μ is a probability measure, and $\mathcal{G}_y$ is measurable since both $\mathcal{G}^+$ and $\mathcal{G}^-$ are closed. 4.12 is then true by definition of a probability measure.

Step 3. *If the conditions of Prop. 13 are satisfied then for all* $(\mathbf{x}, y) \in \mathcal{G}_y$

$$\|\mathbf{x}\|_2 \leq R \tag{4.14}$$

$$\|\mathbf{x}\|_0 \leq s \tag{4.15}$$

$$y\left(\langle \mathbf{w}^\star, \mathbf{x} \rangle + b^\star\right) \geq 1 \tag{4.16}$$

For 4.14, suppose by contradiction that $\exists (\mathbf{x}, y) \in \mathcal{G}^+ \times \{1\}$ or $\exists (\mathbf{x}, y) \in \mathcal{G}^- \times \{-1\}$ such that $\|\mathbf{x}\| = \gamma > R$. Then, since the 2-norm is continuous, there exists $\delta > 0$ such that for all $\dot{\mathbf{x}} \in \mathcal{A} := \{\dot{\mathbf{x}} \in \mathbb{R}^d : \|\mathbf{x} - \dot{\mathbf{x}}\|_2 < \delta\}$, $| \|\mathbf{x}\|_2 - \|\dot{\mathbf{x}}\| | < \gamma - R$. By definition of $\mathcal{G}^+$ and $\mathcal{G}^-$, since $\mathcal{A}$ is open, $\mu(\mathcal{A} \times \{1, -1\}) > 0$ implying $\mathbb{P}_{(\mathbf{x},y)\sim\mu}[\|x\|_2 > R] > 0$ and completing the proof. 4.16 can be proved in a similar manner.

For 4.15, suppose by contradiction that $\exists(\mathbf{x}, y) \in \mathcal{G}^+ \times \{1\}$ or $\exists(\mathbf{x}, y) \in \mathcal{G}^- \times \{-1\}$ such that $\|\mathbf{x}\|_0 > s$. Let $\epsilon = \min_{i \in \text{supp}(\mathbf{x})} \mathbf{x}_i$. Then, $\forall \dot{\mathbf{x}} \in \mathcal{A} := \{\dot{x} : \|\dot{\mathbf{x}} - \mathbf{x}\| < \epsilon\}$, $\|\dot{\mathbf{x}}\|_0 > s$. However, since $\mu(\mathcal{A} \times \{1, -1\}) > 0$ we have a contradiction and complete the proof.

With these propositions we can now prove Lemma. 1. By Eqn 4.16 in lemma. 3, $\forall \mathbf{x} \in \mathcal{G}^+$, $\langle \mathbf{w}^\star, \mathbf{x} \rangle + b^\star \geq 1$ and $\forall \mathbf{x} \in \mathcal{G}^-$, $\langle \mathbf{w}^\star, \mathbf{x} \rangle + b^\star \leq -1$.

For $\mathbf{x} \in \text{co}(\mathcal{G}^+)$, by Caratheodory's theorem, there exists $\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_{d+1} \in \mathcal{G}^+$ and $\alpha \in \mathbb{R}_+^{d+1}$ with $\|\alpha\|_1 = 1$ such that $\mathbf{x} = \sum_{i \in [d+1]} \alpha_i \tilde{\mathbf{x}}_i$. Then, $\langle \mathbf{w}^\star, \mathbf{x} \rangle + b^\star = \sum_{i \in [d+1]} \alpha_i (\langle \mathbf{w}^\star, \tilde{\mathbf{x}}_i \rangle + b^\star) \geq \sum_{i \in [d+1]} \alpha_i = 1$.

Additionally, For $\mathbf{x} \in \text{co}(\mathcal{G}^-)$, by Caratheodory's theorem, there exists $\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_{d+1} \in \mathcal{G}^-$ and $\alpha \in \mathbb{R}_+^{d+1}$ with $\|\alpha\|_1 = 1$ such that $\mathbf{x} = \sum_{i \in [d+1]} \alpha_i \tilde{\mathbf{x}}_i$. Then, $\langle \mathbf{w}^\star, \mathbf{x} \rangle + b^\star = \sum_{i \in [d+1]} \alpha_i (\langle \mathbf{w}^\star, \tilde{\mathbf{x}}_i \rangle + b^\star) \leq \sum_{i \in [d+1]} -\alpha_i = -1$.

Thus, for $\mathbf{x}_1 \in \mathcal{G}^+$ and $\mathbf{x}_2 \in \mathcal{G}^-$, $\langle \mathbf{w}^\star, \mathbf{x}_1 - \mathbf{x}_2 \rangle = (\langle \mathbf{w}^\star, \mathbf{x}_1 \rangle + b^\star) - (\langle \mathbf{w}^\star, \mathbf{x}_2 \rangle + b^\star) \geq 2$

Finally, by the triangle inequality for inner products and the fact $\mathbf{w}^\star$ is not all-zero, we have

$$\|\mathbf{x}_1 - \mathbf{x}_2\| \geq \frac{|\langle \mathbf{w}^\star, \mathbf{x}_1 - \mathbf{x}_2 \rangle|}{\|\mathbf{w}^\star\|} \geq \frac{2}{\|\mathbf{w}^\star\|} \tag{4.17}$$

Therefore there exists $\delta > 0$ and the proof is complete.

We now show that if the distance between the convex hulls of the two supports is bounded away from zero, then we can construct a solution based on the two convex hulls. To do this we need to first show that there exists in the closure of the two convex hulls two points that achieve the infimum.

Step 4. *If $\forall \mathbf{x} \in \mathcal{G}^+$ we have $\|\mathbf{x}\| \leq R$ and $\forall \mathbf{x} \in \mathcal{G}^-$ we have $\|\mathbf{x}\| \leq R$ and*

$$\delta = \inf \{\|\mathbf{x}_1 - \mathbf{x}_2\| : \mathbf{x}_1 \in \text{co}(\mathcal{G}^+), \mathbf{x}_2 \in \text{co}(\mathcal{G}^-)\}$$

Then there exists $\mathbf{x}_1^\star \in \text{cl}(\text{co}(\mathcal{G}^+))$ and $\mathbf{x}_2^\star \in \text{cl}(\text{co}(\mathcal{G}^-))$ such that

$$\|\mathbf{x}_1^\star - \mathbf{x}_2^\star\| = \delta$$

Proof. By definition of infimum, for each $n \in \mathbb{N}$ there exists $\tilde{\mathbf{x}}_n \in \text{co}(\mathcal{G}^+)$ and $\ddot{\mathbf{x}}_n \in \text{co}(\mathcal{G}^-)$ such that $\delta \leq \|\tilde{\mathbf{x}}_n - \ddot{\mathbf{x}}_n\| < \delta + \frac{1}{n}$. Thus

$$\lim_{n\to\infty} \|\tilde{\mathbf{x}}_n - \ddot{\mathbf{x}}_n\| = \delta$$

As $(\tilde{\mathbf{x}}_n)$ is a sequence in $\text{cl}(\text{co}(\mathcal{G}^+))$ and $\text{cl}(\text{co}(\mathcal{G}^+))$ is compact as it is closed and bounded. There exists a subsequence $(\tilde{\mathbf{x}}_{n_k} : k \in \mathbb{N})$ and an element $\bar{\mathbf{x}}$ in $\text{cl}(\text{co}(\mathcal{G}^+))$ such that

$$\lim_{k\to\infty} \|\tilde{\mathbf{x}}_{n_k} - \bar{\mathbf{x}}\| = 0$$

Additionally, as $(\ddot{\mathbf{x}}_{n_k})$ is a sequence in $\text{cl}(\text{co}(\mathcal{G}^-))$ and $\text{cl}(\text{co}(\mathcal{G}^-))$ is compact as it is closed and bounded. There exists a subsequence $(\ddot{\mathbf{x}}_{n_{k_j}} : j \in \mathbb{N})$ and an element $\dot{\mathbf{x}}$ in $\text{cl}(\text{co}(\mathcal{G}^-))$ such that

$$\lim_{j\to\infty} \left\|\ddot{\mathbf{x}}_{n_{k_j}} - \dot{\mathbf{x}}\right\| = 0$$

Then taking these together we have

$$\lim_{j\to\infty} \left\|\tilde{\mathbf{x}}_{n_{k_j}} - \ddot{\mathbf{x}}_{n_{k_j}}\right\| = \|\bar{\mathbf{x}} - \dot{\mathbf{x}}\|$$

which completes the proof.

We can now prove that if the distance between the two convex hulls is bounded away from zero, we can construct a solution that gives linear separability.

Step 5. *If* $\forall \mathbf{x} \in \mathcal{G}^+$ *we have* $\|\mathbf{x}\| \leq R$ *and* $\forall \mathbf{x} \in \mathcal{G}^-$ *we have* $\|\mathbf{x}\| \leq R$ *and if* $\exists \delta > 0$ *such that*

$$\inf \{\|\mathbf{x}_1 - \mathbf{x}_2\| : \mathbf{x}_1 \in \text{co}(\mathcal{G}^+), \mathbf{x}_2 \in \text{co}(\mathcal{G}^-)\} = \delta$$

By step. 7, there exists $\mathbf{x}_1^\star \in \mathrm{cl}(\mathrm{co}(\mathcal{G}^+))$ *and* $\mathbf{x}_2^\star \in \mathrm{cl}(\mathrm{co}(\mathcal{G}^-))$ *such that* $\|\mathbf{x}_1^\star - \mathbf{x}_2^\star\| = \delta$*. Then, with*

$$\mathbf{w}_0 = \frac{2(\mathbf{x}_1^\star - \mathbf{x}_2^\star)}{\|\mathbf{x}_1^\star - \mathbf{x}_2^\star\|^2} \tag{4.18}$$

$$b_0 = 1 - \frac{2\langle \mathbf{x}_1^\star - \mathbf{x}_2^\star, \mathbf{x}_1^\star \rangle}{\|\mathbf{x}_1^\star - \mathbf{x}_2^\star\|^2} \tag{4.19}$$

We have

$$\mathbb{P}_{(\mathbf{x},y)\sim\mu}[y(\langle \mathbf{w}_0, \mathbf{x}\rangle + b_0) \geq 1] = 1$$

Proof. For $\tilde{\mathbf{x}} \in \mathcal{G}^+$

$$\langle \mathbf{w}_0, \tilde{\mathbf{x}}\rangle + b_0 = \left\langle \frac{2(\mathbf{x}_1^\star - \mathbf{x}_2^\star)}{\|\mathbf{x}_1^\star - \mathbf{x}_2^\star\|^2}, \tilde{\mathbf{x}} \right\rangle + 1 - \frac{2\langle \mathbf{x}_1^\star - \mathbf{x}_2^\star, \mathbf{x}_1^\star\rangle}{\|\mathbf{x}_1^\star - \mathbf{x}_2^\star\|^2} =$$

$$= \frac{2\langle \mathbf{x}_1^\star - \mathbf{x}_2^\star, \tilde{\mathbf{x}}\rangle}{\|\mathbf{x}_1^\star - \mathbf{x}_2^\star\|^2} + 1 - \frac{2\langle \mathbf{x}_1^\star - \mathbf{x}_2^\star, \mathbf{x}_1^\star\rangle}{\|\mathbf{x}_1^\star - \mathbf{x}_2^\star\|^2}$$

Thus for $\langle \mathbf{w}_0, \tilde{\mathbf{x}}\rangle + b_0 \geq 1$ we just need

$$\langle \mathbf{x}_1^\star - \mathbf{x}_2^\star, \tilde{\mathbf{x}}\rangle \geq \langle \mathbf{x}_1^\star - \mathbf{x}_2^\star, \mathbf{x}_1^\star\rangle \tag{4.20}$$

By convexity of $\mathrm{cl}(\mathrm{co}(\mathcal{G}^+))$ and by definition of $\mathbf{x}_1^\star$ and $\mathbf{x}_2^\star$ we have for all $\lambda \in (0,1)$. Note that $\lambda \in (0,1)$ ensures $\lambda\tilde{\mathbf{x}} + (1-\lambda)\mathbf{x}_1^\star \in \mathrm{co}(\mathcal{G}^+)$

$$\|\lambda\tilde{\mathbf{x}} + (1-\lambda)\mathbf{x}_1^\star - \mathbf{x}_2^\star\| \geq \|\mathbf{x}_1^\star - \mathbf{x}_2^\star\|$$

Or equivalently

$$\langle \lambda\tilde{\mathbf{x}} + (1-\lambda)\mathbf{x}_1^\star - \mathbf{x}_2^\star, \lambda\tilde{\mathbf{x}} + (1-\lambda)\mathbf{x}_1^\star - \mathbf{x}_2^\star\rangle$$

$$\geq \langle \mathbf{x}_1^\star - \mathbf{x}_2^\star, \mathbf{x}_1^\star - \mathbf{x}_2^\star\rangle$$

Using the linearity of inner products and rearranging we get

$$2\lambda\langle\tilde{\mathbf{x}}-\mathbf{x}_1^\star,\mathbf{x}_1^\star-\mathbf{x}_2^\star\rangle+\lambda^2\langle\tilde{\mathbf{x}}-\lambda\mathbf{x}_1^\star,\lambda\tilde{\mathbf{x}}-\lambda\mathbf{x}_1^\star\rangle\geq 0$$

If we restrict to $\lambda\in(0,1]$ we get

$$\langle\tilde{\mathbf{x}}-\mathbf{x}_1^\star,\mathbf{x}_1^\star-\mathbf{x}_2^\star\rangle\geq\frac{-\lambda\left\|\tilde{\mathbf{x}}-\mathbf{x}_1^\star\right\|^2}{2}$$

And

$$\langle\tilde{\mathbf{x}}-\mathbf{x}_1^\star,\mathbf{x}_1^\star-\mathbf{x}_2^\star\rangle\geq\sup_{\lambda\in(0,1]}\frac{-\lambda\left\|\tilde{\mathbf{x}}-\mathbf{x}_1^\star\right\|^2}{2}=0$$

Thus,$\langle\tilde{\mathbf{x}},\mathbf{x}_1^\star-\mathbf{x}_2^\star\rangle\geq\langle\mathbf{x}_1^\star,\mathbf{x}_1^\star-\mathbf{x}_2^\star\rangle$ which shows 5.5

Similarly, for $\langle\mathbf{w}_0,\tilde{\mathbf{x}}\rangle+b_0\leq-1$ for $\tilde{\mathbf{x}}\in\mathcal{G}^-$ we need

$$2\langle\mathbf{x}_1^\star-\mathbf{x}_2^\star,\tilde{\mathbf{x}}\rangle-2\langle\mathbf{x}_1^\star-\mathbf{x}_2^\star,\mathbf{x}_1^\star\rangle\leq-2\left\|\mathbf{x}_1^\star-\mathbf{x}_2^\star\right\|^2$$

Or equivalently,

$$\langle\mathbf{x}_1^\star-\mathbf{x}_2^\star,\mathbf{x}_2^\star\rangle\geq\langle\mathbf{x}_1^\star-\mathbf{x}_2^\star,\tilde{\mathbf{x}}\rangle$$

This is true by the same convexity argument used previously. Thus, we have $y(\langle\mathbf{w}_0,\mathbf{x}\rangle+b)$ is true for all $(\mathbf{x},y)\in\mathcal{G}_y$. By step. 2, $\mathbb{P}_\mu[(\mathbf{x},y)\in\mathcal{G}_y]=1$ which completes the proof.

We will now show that the solution defined in equations 5.3 and 5.4 satisfies the compressed inner product property we desire.

Step 6. *If $\mathbf{w}_0,b_0$ are defined by 5.3 and 5.4, and the compression matrix Φ has $2s$-restricted isometric constant δ_{2s}. Then, for all s-sparse $\mathbf{x}\in\mathbf{R}^d$ such that $\|\mathbf{x}\|\leq R$*

$$\left|\langle\Phi\mathbf{w}_0,\Phi\mathbf{x}\rangle-\langle\mathbf{w}_0,\mathbf{x}\rangle\right|\leq\delta_{2s}\left\|\mathbf{w}_0\right\|^2R^2$$

Proof. There exists $\tilde{\mathbf{x}}^1, \tilde{\mathbf{x}}^2, \tilde{\mathbf{x}}^3, \ldots \in \mathrm{co}(\mathcal{G}^+)$ such that $\mathbf{x}_1^\star = \lim_{n\to\infty} \tilde{\mathbf{x}}^n$. Similarly, there exists $\ddot{\mathbf{x}}^1, \ddot{\mathbf{x}}^2, \ldots \in \mathrm{co}(\mathcal{G}^-)$ such that $\mathbf{x}_2^\star = \lim_{n\to\infty} \ddot{\mathbf{x}}^n$.

Since $\tilde{\mathbf{x}}^j \in \mathcal{G}^+$, by Caratheodory's theorem, there exists $\tilde{\mathbf{x}}_1^j, \tilde{\mathbf{x}}_2^j, \ldots, \tilde{\mathbf{x}}_{d+1}^j \in \mathcal{G}^+$ and $\boldsymbol{\alpha}^j \in \mathbb{R}_+^{d+1}$ with $\|\boldsymbol{\alpha}^j\|_1 = 1$ such that

$$\tilde{\mathbf{x}}^j = \sum_{n\in[d+1]} \boldsymbol{\alpha}_n^j \tilde{\mathbf{x}}_n^j$$

Similarly for $\ddot{\mathbf{x}}^j \in \mathcal{G}^-$, there exists $\ddot{\mathbf{x}}_1^j, \ddot{\mathbf{x}}_2^j, \ldots, \ddot{\mathbf{x}}_{d+1}^j \in \mathcal{G}^-$ and $\boldsymbol{\beta}^j \in \mathbb{R}_+^{d+1}$ with $\|\boldsymbol{\beta}^j\|_1 = 1$ such that

$$\ddot{\mathbf{x}}^j = \sum_{n\in[d+1]} \boldsymbol{\beta}_n^j \ddot{\mathbf{x}}_n^j$$

For $j \in \mathbb{N}$, let

$$\mathbf{w}_0^j = \frac{2(\tilde{\mathbf{x}}^j - \ddot{\mathbf{x}}^j)}{\|\tilde{\mathbf{x}}^j - \ddot{\mathbf{x}}^j\|^2}$$

Then,

$$\langle \Phi \mathbf{w}_0^j, \Phi \mathbf{x}\rangle = \frac{2}{\|\tilde{\mathbf{x}}^j - \ddot{\mathbf{x}}^j\|^2} \langle \Phi(\tilde{\mathbf{x}}^j - \ddot{\mathbf{x}}^j), \Phi \mathbf{x}\rangle$$

By the linearity of inner products,

$$= \frac{2}{\|\tilde{\mathbf{x}}^j - \ddot{\mathbf{x}}^j\|^2} \left(\sum_{n\in[d+1]} \boldsymbol{\alpha}_n^j \left\langle \Phi \tilde{\mathbf{x}}_n^j, \Phi \mathbf{x} \right\rangle - \sum_{n\in[d+1]} \boldsymbol{\beta}_n^j \left\langle \Phi \ddot{\mathbf{x}}_n^j, \Phi \mathbf{x} \right\rangle \right)$$

$$\leq \frac{2}{\|\tilde{\mathbf{x}}^j - \ddot{\mathbf{x}}^j\|^2} \left(\sum_{n\in[d+1]} \boldsymbol{\alpha}_n^j (\left\langle \tilde{\mathbf{x}}_n^j, \mathbf{x} \right\rangle + \delta_{2s} R^2) - \sum_{n\in[d+1]} \boldsymbol{\beta}_n^j (\left\langle \ddot{\mathbf{x}}_n^j, \mathbf{x} \right\rangle - \delta_{2s} R^2) \right) \quad \text{(a)}$$

$$\leq \langle \mathbf{w}_0^j, \mathbf{x} \rangle + \delta_{2s} \left\| \mathbf{w}_0^j \right\| R^2 \qquad \text{(a)}$$

Where (a) follows from using lemma 11 and (b) by rearranging

We can repeat the process to get the lower bound. Combining the two we get

$$|\langle \Phi \mathbf{w}_0^j, \Phi \mathbf{x} \rangle - \langle \mathbf{w}_0^j, \mathbf{x} \rangle| \leq \delta_{2s} \left\| \mathbf{w}_0^j \right\| R^2$$

Using the fact that if $\lim_{n\to\infty} \mathbf{c}_n = \mathbf{c}$ we have $\lim_{n\to\infty} \langle \mathbf{c}_n, \mathbf{x} \rangle = \langle \mathbf{c}, \mathbf{x} \rangle$ and $\lim_{n\to\infty} \|\mathbf{c}_n\| = \|\mathbf{c}\|$ we take the limit of each side and get

$$|\langle \Phi \mathbf{w}_0, \Phi \mathbf{x} \rangle - \langle \mathbf{w}_0, \mathbf{x} \rangle| \leq \delta_{2s} \|\mathbf{w}_0\| R^2$$

which completes the proof.

We now have the necessary results to prove Prop. 13. By step. 1 we have the distance between the two supports is bounded by δ. By step. 8 we then have another solution $\mathbf{w}_0, b_0$ defined by the two supports. By step 9 we have the compressed inner product deviation property. Finally we show that $\|\mathbf{w}_0\| \leq \|\mathbf{w}^\star\|$. By the properties of closures and limits, Eqn 5.2 also holds for all $\mathbf{x}_1 \in \text{cl}(\text{co}(\mathcal{G}^+))$ and $\mathbf{x}_2 \in \text{cl}(\text{co}(\mathcal{G}^-))$. Thus, $\|\mathbf{w}^\star\| \geq \frac{2}{\|\mathbf{x}_1^\star - \mathbf{x}_2^\star\|} = \|\mathbf{w}_0\|$ which completes the proof.

4.7 Proof of Main Results

4.7.1 Proof of Thm. 9

Since the conditions of Prop. 13 are satisfied we have $\mathbf{w}_0, b^\star$ are linearly separable solutions. By lemma. 3 and Eqn. 4.11, we have for all $(\mathbf{x}, y) \in \mathcal{G}_y$

$$y(\langle \Phi \mathbf{w}_0, \Phi \mathbf{x} \rangle + b^\star) \geq 1 - \delta_{2s} \|\mathbf{w}_0\|^2 R^2$$

If $\delta_{2s} < \frac{1}{\|\mathbf{w}^\star\|^2 R^2}$, $\delta_{2s} \|\mathbf{w}_0\|^2 R^2 < \frac{\|\mathbf{w}_0\|^2 R^2}{\|\mathbf{w}^\star\|^2 R^2} \leq \frac{\|\mathbf{w}^\star\|^2 R^2}{\|\mathbf{w}^\star\|^2 R^2} = 1$ by Thm. 13. and

$$\frac{y}{1 - \delta_{2s} \|\mathbf{w}_0\|^2 R^2} (\langle \Phi \mathbf{w}_0, \Phi \mathbf{x} \rangle + b^\star) \geq 1$$

Then,

$$= y \left(\left\langle \Phi \frac{\mathbf{w}_0}{1 - \delta_{2s} \|\mathbf{w}_0\|^2 R^2}, \Phi \mathbf{x} \right\rangle + \frac{b^\star}{1 - \delta_{2s} \|\mathbf{w}_0\|^2 R^2} \right) \geq 1$$

Thus, $\mathbf{w}_C = \frac{\mathbf{w}_0}{1-\delta_{2s}\|\mathbf{w}_0\|^2 R^2}$ and $b_C = \frac{b^\star}{1-\delta_{2s}\|\mathbf{w}_0\|^2 R^2}$ are linearly separable for all $(\mathbf{x}, y) \in \mathcal{G}_y$. By lemma. 2 this is then true with probability 1 in the compressed domain completing the proof.

4.7.2 Proof of Thm. 10

Let $\mathbf{x}_1, \mathbf{x}_2$ be defined by Eqn. 4.3 in the uncompressed domain. Let $\mathbf{z}_1, \mathbf{z}_2$ be defined by Eqn. 4.3 in the compressed domain. Let $\mathbf{x'}_1 \in \text{co}(\mathcal{S}^1)$ s.t. $\Phi \mathbf{x'}_1 = \mathbf{z}_1$ be uncompressed x-value of the training point that was compressed to $\mathbf{z}_1$. Similarly for $\mathbf{x'}_2 \in \text{co}(\mathcal{S}^{-1})$ s.t. $\Phi \mathbf{x'}_2 = \mathbf{z}_2$. Then, by lemma 12

$$\left\|\mathbf{w}_{\mathcal{S}}^{\Phi}\right\|^2 = \frac{4}{\|\mathbf{z}_1 - \mathbf{z}_2\|^2} \leq \frac{4}{\|\mathbf{x'}_1 - \mathbf{x'}_2\|^2 - 4\delta_{2s} R^2}$$

By definition, $\mathbf{x}_1, \mathbf{x}_2$ achieves the minimum so

$$\leq \frac{4}{\|\mathbf{x}_1 - \mathbf{x}_2\|^2 - 4\delta_{2s} R^2} = \frac{4}{\frac{4}{\|\mathbf{w}_{\mathcal{S}}\|^2} - 4\delta_{2s} R^2}$$

$$= \frac{\|\mathbf{w}_{\mathcal{S}}\|^2}{1 - \delta_{2s} R^2 \|\mathbf{w}_{\mathcal{S}}\|^2} \tag{4.21}$$

Since the compressed domain is linearly separable, Thm. 8 applies and we get

$$\mathop{\mathbb{P}}_{(\mathbf{x},y)\sim\mu}[y \neq \text{sign}(\langle \mathbf{w}_{\mathcal{S}}^{\Phi}, \Phi\mathbf{x}\rangle + b_{\mathcal{S}}^{\Phi})] \leq \frac{8R||\mathbf{w}_{\mathcal{S}}^{\Phi}|| + 2}{\sqrt{m}} + \sqrt{\frac{\ln\left(\frac{4\log_2\left(||\mathbf{w}_{\mathcal{S}}^{\Phi}||\right)}{\delta}\right)}{m}} \quad (4.22)$$

Since $\|\mathbf{w}_{\mathcal{S}}\| \leq \|\mathbf{w}^{\star}\|$ and by assumption $\delta_{2s} < \frac{1}{\|\mathbf{w}^{\star}\|^2 R^2}$ we have $\delta_{2s} R^2 \|\mathbf{w}_{\mathcal{S}}\|^2 < 1$. Additionally, as $\delta_{2s}, R, \|.\|$ are positive we get $0 \leq \delta_{2s} R^2 \|\mathbf{w}_{\mathcal{S}}\|^2 < 1$ and thus $\frac{\|\mathbf{w}_{\mathcal{S}}\|^2}{1-\delta_{2s}R^2\|\mathbf{w}_{\mathcal{S}}\|^2} \geq \|\mathbf{w}_{\mathcal{S}}\|^2$. Then, plugging Eqn. 4.21 into Eqn. 4.22 and using the fact that each term in Eqn. 4.22 is concave with respect to $\|\mathbf{w}_{\mathcal{S}}\|$ we get

$$\mathop{\mathbb{P}}_{(\mathbf{x},y)\sim\mu}[y \neq \text{sign}(\langle \mathbf{w}_{\mathcal{S}}^{\Phi}, \Phi\mathbf{x}\rangle + b_{\mathcal{S}}^{\Phi})] \leq \frac{\mathcal{L}_B}{\sqrt{1 - \delta_{2s} R^2 \|\mathbf{w}_{\mathcal{S}}\|^2}}$$

And finally, since $\|\mathbf{w}_{\mathcal{S}}\| \leq \|\mathbf{w}^{\star}\|$ we get

$$\leq \frac{\mathcal{L}_B}{\sqrt{1 - \delta_{2s} R^2 \|\mathbf{w}^{\star}\|^2}}$$

which completes the proof.

4.8 Conclusion

In this chapter, we have shown that compressed learning for hard-SVM is possible. We showed that if the restricted isometric constant of the compression matrix is bounded by $\frac{1}{\|\mathbf{w}^{\star}\|R^2}$ the separability assumption holds. Additionally, we showed after compression, the generalization bounds increases as $\frac{\mathcal{L}_B}{\sqrt{1-\delta_{2s}R^2\|\mathbf{w}^{\star}\|^2}}$.

5. COMPRESSED LEARNING WITH HARD-SVM WITHOUT SPARSITY

5.1 Introduction

In this chapter we analyze compressed learning when we no longer have a sparsity assumption. This allows for a more general analysis but restricts the compression amount.

5.2 Background

We rely on the result from the Johnson-Lindenstrauss Lemma for infinite sets. This result allows us to bound the pairwise distance after compression in terms of the Guassian width of the set.

Theorem 14 (Additive Johnson-Lindenstrauss Lemma for Infinite Sets)**.** *Consider a set $\mathcal{X} \subset \mathbb{R}^n$. Let A be an $m \times n$ matrix whose rows A_i are independent, isotropic, sub-gaussian random vectors in $\mathbb{R}^n$. Then, with high probability (say, 0.99), the scaled matrix*

$$Q = \frac{1}{\sqrt{m}} A$$

satisfies

$$\|x - y\|_2 - \chi \leq \|Qx - Qy\|_2 \leq \|x - y\|_2 + \chi$$

for all $x, y \in \mathcal{X}$. Where

$$\chi = \frac{CK^2 w(\mathcal{X})}{\sqrt{m}}$$

and $K = \max_i \|A_i\|_{\Psi_2}$

The set we care about for this analysis is the combined support. That is, the combined support is $\mathcal{G} = \mathcal{G}^+ \cup \mathcal{G}^-$. The gaussian width is defined

$$w(\mathcal{G}) = \mathbb{E}_g \sup_{\mathbf{x} \in \mathcal{G}} \langle g, \mathbf{x} \rangle$$

We assume that $\vec{0}$ is contained in the convex hull of the combined support. This is a very reasonable assumption as the normalization is a coming part of data preprocessing. This also allows us the ability to bound the inner product of two vectors in the combined support.

Lemma 15. *The inner product of* $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{G}$ *satisfies*

$$|\langle Q\mathbf{x}_1, Q\mathbf{x}_2\rangle - \langle \mathbf{x}_1, \mathbf{x}_2\rangle| \leq 3\chi'$$

Proof: By definition,

$$\langle \mathbf{x}_1, \mathbf{x}_2\rangle = \frac{\|\mathbf{x}_1\|^2 + \|\mathbf{x}_2\|^2 - \|\mathbf{x}_1 - \mathbf{x}_2\|^2}{2}$$

By plugging this in

$$|\langle Q\mathbf{x}_1, Q\mathbf{x}_2\rangle - \langle \mathbf{x}_1, \mathbf{x}_2\rangle|$$

$$= \frac{1}{2}\left|\left(\left(\|Q\mathbf{x}_1\|^2 - \|\mathbf{x}_1\|^2\right) + \left(\|Q\mathbf{x}_2\|^2 - \|\mathbf{x}_2\|^2\right) - \left(\|Q\mathbf{x}_1 - Q\mathbf{x}_2\|^2 - \|\mathbf{x}_1 - \mathbf{x}_2\|^2\right)\right)\right|$$

$$\leq \frac{3}{2}\chi'$$

where

$$\chi' = 4R\chi + \chi^2$$

making the following true.

$$\|x - y\|_2^2 - \chi' \leq \|Qx - Qy\|_2^2 \leq \|x - y\|_2^2 + \chi'$$

for all $x, y \in \mathcal{X}$

Lemma 16. *The Guassian width* $w(\mathcal{G}) = w(\text{cl}(\text{co}(\mathcal{G})))$

Proof. By Vershynin [20], $w(\mathcal{G}) = w(\text{co}(\mathcal{G}))$ so we just need to show $w(\text{cl}(\text{co}(\mathcal{G}))) = w(\text{co}(\mathcal{G}))$.

For a fixed vector g, $\sup_{x \in \text{co}(\mathcal{G})} \langle g, \mathbf{x} \rangle < \sup_{x \in \text{cl}(\text{co}(\mathcal{G}))} \langle g, \mathbf{x} \rangle$ by the fact that $\text{co}(\mathcal{G}) \subset \text{cl}(\text{co}(\mathcal{G}))$.

Then, for every point $\mathbf{x} \in \text{cl}(\text{co}(\mathcal{G}))$, there exists at least one sequence $\mathbf{x}_1, \mathbf{x}_2, \ldots \subset \text{co}(\mathcal{G})$ such that $\lim_{n \to \infty} \mathbf{x}_n = \mathbf{x}$. Since for all n, $\langle \mathbf{x}_n, g \rangle \leq \sup_{x \in \text{co}(\mathcal{G})} \langle g, \mathbf{x} \rangle$, we have $\langle \mathbf{x}, g \rangle \leq \sup_{x \in \text{co}(\mathcal{G})} \langle g, \mathbf{x} \rangle$. Since the choice of $\mathbf{x} \in \text{cl}(\text{co}(\mathcal{G}))$ was arbitrary, this is true for all $x \in \text{cl}(\text{co}(\mathcal{G}))$

This completes the proof

5.2.1 Constants

Before going into tho resulting theorem, we will first discuss the constant K in the infinite set JL lemma. In the theorem statement, K is defined as the $K = \max_i \|A_i\|_{\Psi_2}$ where $\|.\|_{\Psi_2}$ is the sub-gaussian norm.

The sub-gaussian norm for a random variable X is defined

$$\|X\|_{\Psi_2} = \inf \left\{ t > 0 : \ \mathbb{E} \exp(X^2/t^2) \leq 2 \right\}$$

An example is if $X \sim N(0,1)$ is distributed according to a normal distribution. Then the $\|X\|_{\Psi_2} \leq C$ is bounded by an absolute constant C.

The sub-gaussian norm for a random vector $X \in \mathbb{R}^n$ is the suprema over the marginal sub-gaussian norms in any possible direction. More formally, it is defined

$$\|X\|_{\Psi_2} = \sup_{x \in S^{n-1}} \|\langle X, x \rangle\|_{\Psi_2}$$

where S^{n-1} is the unit sphere. If the vector has iid entries drawn $N(0,1)$ then the every direction's marginal distribution has the same sub-gaussian norm and the sub-gaussian norm of the vector is bounded by a constant C.

This means if we create our matrix A, from the infinite set JL lemma, by sampling each entry A_{ij} from $N(0,1)$ then K would be bounded by a constant C.

5.3 Result

We now present the main result of this section.

Theorem 17. *If μ satisfies $\mathbb{P}_{(\mathbf{x},y)\sim\mu}[\|x\|_2 \leq R] = 1$ and if $\exists \mathbf{w}^\star, b^\star$ such that $\mathbb{P}_{(\mathbf{x},y)\sim\mu}[y\left(\langle \mathbf{w}^\star, \mathbf{x}\rangle + b^\star\right) \geq 1] = 1$. Then with m such that $3\left\|\mathbf{w}^\star\right\|^2 (4R(\frac{CK^2w(\mathcal{X})}{\sqrt{m}}) + (\frac{CK^2w(\mathcal{X})}{\sqrt{m}})^2) < 1$ let A be an $m \times n$ matrix whose rows A_i are independent, isotropic, sub-gaussian random vectors in $\mathbb{R}^n$. Then, with high probability (say, 0.99), the scaled matrix $Q = \frac{1}{\sqrt{m}}A$ maps the domain to a linearly separable compressed domain. That is, $\exists \mathbf{w}_C \in \mathbb{R}^l, b_C \in \mathbb{R}$ such that*

$$\mathbb{P}_{(\mathbf{x},y)\sim\mu}[y(\langle \mathbf{w}_C, Q\mathbf{x}\rangle + b_C) \geq 1] = 1$$

To prove this theorem, we follow the same steps as the compressed setting. We first construct a compressable solution that have properties we can exploit after compression.

Proposition 18. *If μ satisfies $\mathbb{P}_{(\mathbf{x},y)\sim\mu}[\|x\|_2 \leq R] = 1$ and if $\exists \mathbf{w}^\star, b^\star$ such that $\mathbb{P}_{(\mathbf{x},y)\sim\mu}[y\left(\langle \mathbf{w}^\star, \mathbf{x}\rangle + b^\star\right) \geq 1] = 1$. Then, there exist $\mathbf{w}_0$ such that $\mathbb{P}_{(\mathbf{x},y)\sim\mu}[y\left(\langle \mathbf{w_0}, \mathbf{x}\rangle + b^\star\right) \geq 1] = 1$, $\|\mathbf{w}_0\| \leq \|\mathbf{w}^\star\|$ and for all $\mathbf{x} \in \mathcal{G}$*

$$|\langle Q\mathbf{w}_0, Q\mathbf{x}\rangle - \langle \mathbf{w}_0, \mathbf{x}\rangle| \leq 3\left\|\mathbf{w}_0\right\|^2 \chi' \tag{5.1}$$

5.4 Proof

To prove this result, we first show that the distance between the two convex hulls in a linear separable uncompressed domain is bounded away from zero.

Lemma 19. *If μ satisfies $\mathbb{P}_{(\mathbf{x},y)\sim\mu}[\|x\|_2 \leq R] = 1$ and if $\exists \mathbf{w}^\star, b^\star$ such that $\mathbb{P}_{(\mathbf{x},y)\sim\mu}[y\left(\langle \mathbf{w}^\star, \mathbf{x}\rangle + b^\star\right) \geq 1] = 1$. Then, $\exists \delta > 0$ such that*

$$\inf\left\{\|\mathbf{x}_1 - \mathbf{x}_2\| : \mathbf{x}_1 \in \text{co}(\mathcal{G}^+), \mathbf{x}_2 \in \text{co}(\mathcal{G}^-)\right\} = \delta$$

By Eqn 4.16 in lemma. 3, $\forall \mathbf{x} \in \mathcal{G}^+$, $\langle \mathbf{w}^\star, \mathbf{x}\rangle + b^\star \geq 1$ and $\forall \mathbf{x} \in \mathcal{G}^-$, $\langle \mathbf{w}^\star, \mathbf{x}\rangle + b^\star \leq -1$.

For $\mathbf{x} \in \text{co}(\mathcal{G}^+)$, by Caratheodory's theorem, there exists $\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_{d+1} \in \mathcal{G}^+$ and $\alpha \in \mathbb{R}_+^{d+1}$ with $\|\alpha\|_1 = 1$ such that $\mathbf{x} = \sum_{i\in[d+1]} \alpha_i \tilde{\mathbf{x}}_i$. Then, $\langle \mathbf{w}^\star, \mathbf{x}\rangle + b^\star = \sum_{i\in[d+1]} \alpha_i(\langle \mathbf{w}^\star, \tilde{\mathbf{x}}_i\rangle + b^\star) \geq \sum_{i\in[d+1]} \alpha_i = 1$.

Additionally, For $\mathbf{x} \in \text{co}(\mathcal{G}^-)$, by Caratheodory's theorem, there exists $\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_{d+1} \in \mathcal{G}^-$ and $\alpha \in \mathbb{R}_+^{d+1}$ with $\|\alpha\|_1 = 1$ such that $\mathbf{x} = \sum_{i\in[d+1]} \alpha_i \tilde{\mathbf{x}}_i$. Then, $\langle \mathbf{w}^\star, \mathbf{x}\rangle + b^\star = \sum_{i\in[d+1]} \alpha_i(\langle \mathbf{w}^\star, \tilde{\mathbf{x}}_i\rangle + b^\star) \leq \sum_{i\in[d+1]} -\alpha_i = -1$.

Thus, for $\mathbf{x}_1 \in \mathcal{G}^+$ and $\mathbf{x}_2 \in \mathcal{G}^-$, $\langle \mathbf{w}^\star, \mathbf{x}_1 - \mathbf{x}_2\rangle = (\langle \mathbf{w}^\star, \mathbf{x}_1\rangle + b^\star) - (\langle \mathbf{w}^\star, \mathbf{x}_2\rangle + b^\star) \geq 2$

Finally, by the triangle inequality for inner products and the fact $\mathbf{w}^\star$ is not all-zero, we have

$$\|\mathbf{x}_1 - \mathbf{x}_2\| \geq \frac{|\langle \mathbf{w}^\star, \mathbf{x}_1 - \mathbf{x}_2\rangle|}{\|\mathbf{w}^\star\|} \geq \frac{2}{\|\mathbf{w}^\star\|} \tag{5.2}$$

Step 7. *If* $\forall \mathbf{x} \in \mathcal{G}^+$ *we have* $\|\mathbf{x}\| \leq R$ *and* $\forall \mathbf{x} \in \mathcal{G}^-$ *we have* $\|\mathbf{x}\| \leq R$ *and*

$$\delta = \inf \{\|\mathbf{x}_1 - \mathbf{x}_2\| : \mathbf{x}_1 \in \text{co}(\mathcal{G}^+), \mathbf{x}_2 \in \text{co}(\mathcal{G}^-)\}$$

Then there exists $\mathbf{x}_1^\star \in \text{cl}(\text{co}(\mathcal{G}^+))$ *and* $\mathbf{x}_2^\star \in \text{cl}(\text{co}(\mathcal{G}^-))$ *such that*

$$\|\mathbf{x}_1^\star - \mathbf{x}_2^\star\| = \delta$$

Proof. By definition of infimum, for each $n \in \mathbb{N}$ there exists $\tilde{\mathbf{x}}_n \in \text{co}(\mathcal{G}^+)$ and $\ddot{\mathbf{x}}_n \in \text{co}(\mathcal{G}^-)$ such that $\delta \leq \|\tilde{\mathbf{x}}_n - \ddot{\mathbf{x}}_n\| < \delta + \frac{1}{n}$. Thus

$$\lim_{n\to\infty} \|\tilde{\mathbf{x}}_n - \ddot{\mathbf{x}}_n\| = \delta$$

As $(\tilde{\mathbf{x}}_n)$ is a sequence in $\text{cl}(\text{co}(\mathcal{G}^+))$ and $\text{cl}(\text{co}(\mathcal{G}^+))$ is compact as it is closed and bounded. There exists a subsequence $(\tilde{\mathbf{x}}_{n_k} : k \in \mathbb{N})$ and an element $\bar{\mathbf{x}}$ in $\text{cl}(\text{co}(\mathcal{G}^+))$ such that

$$\lim_{k\to\infty} \|\tilde{\mathbf{x}}_{n_k} - \bar{\mathbf{x}}\| = 0$$

Additionally, as $(\ddot{\mathbf{x}}_{n_k})$ is a sequence in $\text{cl}(\text{co}(\mathcal{G}^-))$ and $\text{cl}(\text{co}(\mathcal{G}^-))$ is compact as it is closed

and bounded. There exists a subsequence $(\ddot{\mathbf{x}}_{n_{k_j}} : j \in \mathbb{N})$ and an element $\dot{\mathbf{x}}$ in $\mathrm{cl}(\mathrm{co}(\mathcal{G}^-))$ such that

$$\lim_{j\to\infty} \left\| \ddot{\mathbf{x}}_{n_{k_j}} - \dot{\mathbf{x}} \right\| = 0$$

Then taking these together we have

$$\lim_{j\to\infty} \left\| \tilde{\mathbf{x}}_{n_{k_j}} - \ddot{\mathbf{x}}_{n_{k_j}} \right\| = \|\bar{\mathbf{x}} - \dot{\mathbf{x}}\|$$

which completes the proof.

Step 8. *If* $\forall \mathbf{x} \in \mathcal{G}^+$ *we have* $\|\mathbf{x}\| \leq R$ *and* $\forall \mathbf{x} \in \mathcal{G}^-$ *we have* $\|\mathbf{x}\| \leq R$ *and if* $\exists \delta > 0$ *such that*

$$\inf \{ \|\mathbf{x}_1 - \mathbf{x}_2\| : \mathbf{x}_1 \in \mathrm{co}(\mathcal{G}^+), \mathbf{x}_2 \in \mathrm{co}(\mathcal{G}^-) \} = \delta$$

By step. 7, there exists $\mathbf{x}_1^\star \in \mathrm{cl}(\mathrm{co}(\mathcal{G}^+))$ *and* $\mathbf{x}_2^\star \in \mathrm{cl}(\mathrm{co}(\mathcal{G}^-))$ *such that* $\|\mathbf{x}_1^\star - \mathbf{x}_2^\star\| = \delta$*. Then, with*

$$\mathbf{w}_0 = \frac{2(\mathbf{x}_1^\star - \mathbf{x}_2^\star)}{\|\mathbf{x}_1^\star - \mathbf{x}_2^\star\|^2} \tag{5.3}$$

$$b_0 = 1 - \frac{2\langle \mathbf{x}_1^\star - \mathbf{x}_2^\star, \mathbf{x}_1^\star \rangle}{\|\mathbf{x}_1^\star - \mathbf{x}_2^\star\|^2} \tag{5.4}$$

We have

$$\mathbb{P}_{(\mathbf{x},y)\sim\mu}[y(\langle \mathbf{w}_0, \mathbf{x} \rangle + b_0) \geq 1] = 1$$

Proof. For $\tilde{\mathbf{x}} \in \mathcal{G}^+$

$$\langle \mathbf{w}_0, \tilde{\mathbf{x}} \rangle + b_0 = \left\langle \frac{2(\mathbf{x}_1^\star - \mathbf{x}_2^\star)}{\|\mathbf{x}_1^\star - \mathbf{x}_2^\star\|^2}, \tilde{\mathbf{x}} \right\rangle + 1 - \frac{2\langle \mathbf{x}_1^\star - \mathbf{x}_2^\star, \mathbf{x}_1^\star \rangle}{\|\mathbf{x}_1^\star - \mathbf{x}_2^\star\|^2} =$$

$$= \frac{2\langle \mathbf{x}_1^\star - \mathbf{x}_2^\star, \tilde{\mathbf{x}} \rangle}{\|\mathbf{x}_1^\star - \mathbf{x}_2^\star\|^2} + 1 - \frac{2\langle \mathbf{x}_1^\star - \mathbf{x}_2^\star, \mathbf{x}_1^\star \rangle}{\|\mathbf{x}_1^\star - \mathbf{x}_2^\star\|^2}$$

Thus for $\langle \mathbf{w}_0, \tilde{\mathbf{x}} \rangle + b_0 \geq 1$ we just need

$$\langle \mathbf{x}_1^\star - \mathbf{x}_2^\star, \tilde{\mathbf{x}} \rangle \geq \langle \mathbf{x}_1^\star - \mathbf{x}_2^\star, \mathbf{x}_1^\star \rangle \tag{5.5}$$

By convexity of $\text{cl}(\text{co}(\mathcal{G}^+))$ and by definition of $\mathbf{x}_1^\star$ and $\mathbf{x}_2^\star$ we have for all $\lambda \in (0,1)$. Note that $\lambda \in (0,1)$ ensures $\lambda\tilde{\mathbf{x}} + (1-\lambda)\mathbf{x}_1^\star \in \text{co}(\mathcal{G}^+)$

$$\|\lambda\tilde{\mathbf{x}} + (1-\lambda)\mathbf{x}_1^\star - \mathbf{x}_2^\star\| \geq \|\mathbf{x}_1^\star - \mathbf{x}_2^\star\|$$

Or equivalently

$$\langle \lambda\tilde{\mathbf{x}} + (1-\lambda)\mathbf{x}_1^\star - \mathbf{x}_2^\star, \lambda\tilde{\mathbf{x}} + (1-\lambda)\mathbf{x}_1^\star - \mathbf{x}_2^\star \rangle \geq \langle \mathbf{x}_1^\star - \mathbf{x}_2^\star, \mathbf{x}_1^\star - \mathbf{x}_2^\star \rangle$$

Using the linearity of inner products and rearranging we get

$$2\lambda\langle \tilde{\mathbf{x}} - \mathbf{x}_1^\star, \mathbf{x}_1^\star - \mathbf{x}_2^\star \rangle + \lambda^2 \langle \tilde{\mathbf{x}} - \lambda\mathbf{x}_1^\star, \lambda\tilde{\mathbf{x}} - \lambda\mathbf{x}_1^\star \rangle \geq 0$$

If we restrict to $\lambda \in (0,1]$ we get

$$\langle \tilde{\mathbf{x}} - \mathbf{x}_1^\star, \mathbf{x}_1^\star - \mathbf{x}_2^\star \rangle \geq \frac{-\lambda \|\tilde{\mathbf{x}} - \mathbf{x}_1^\star\|^2}{2}$$

And

$$\langle \tilde{\mathbf{x}} - \mathbf{x}_1^\star, \mathbf{x}_1^\star - \mathbf{x}_2^\star \rangle \geq \sup_{\lambda \in (0,1]} \frac{-\lambda \|\tilde{\mathbf{x}} - \mathbf{x}_1^\star\|^2}{2} = 0$$

Thus, $\langle \tilde{\mathbf{x}}, \mathbf{x}_1^\star - \mathbf{x}_2^\star \rangle \geq \langle \mathbf{x}_1^\star, \mathbf{x}_1^\star - \mathbf{x}_2^\star \rangle$ which shows 5.5

Similarly, for $\langle \mathbf{w}_0, \tilde{\mathbf{x}} \rangle + b_0 \leq -1$ for $\tilde{\mathbf{x}} \in \mathcal{G}^-$ we need

$$2\langle \mathbf{x}_1^\star - \mathbf{x}_2^\star, \tilde{\mathbf{x}} \rangle - 2\langle \mathbf{x}_1^\star - \mathbf{x}_2^\star, \mathbf{x}_1^\star \rangle \leq -2\|\mathbf{x}_1^\star - \mathbf{x}_2^\star\|^2$$

Or equivalently,

$$\langle \mathbf{x}_1^\star - \mathbf{x}_2^\star, \mathbf{x}_2^\star \rangle \geq \langle \mathbf{x}_1^\star - \mathbf{x}_2^\star, \tilde{\mathbf{x}} \rangle$$

This is true by the same convexity argument used previously. Thus, we have $y(\langle \mathbf{w}_0, \mathbf{x} \rangle + b)$ is true for all $(\mathbf{x}, y) \in \mathcal{G}_y$. By step. 2, $\mathbb{P}_\mu[(\mathbf{x}, y) \in \mathcal{G}_y] = 1$ which completes the proof.

We will now show that the solution defined in equations 5.3 and 5.4 satisfies the compressed inner product property we desire.

Step 9. *If $\mathbf{w}_0, b_0$ are defined by 5.3 and 5.4. Then, for all $\mathbf{x} \in \mathcal{G}$*

$$|\langle Q\mathbf{w}_0, Q\mathbf{x} \rangle - \langle \mathbf{w}_0, \mathbf{x} \rangle| \leq 3\chi' \left\| \mathbf{w}_0 \right\|^2$$

Proof. By definition, $\mathbf{x}_1^\star, \mathbf{x}_2^\star \in \text{cl}(\text{co}(\mathcal{G}))$. And,

$$\mathbf{w}_0 = \frac{2(\mathbf{x}_1^\star - \mathbf{x}_2^\star)}{\left\| \mathbf{x}_1^\star - \mathbf{x}_2^\star \right\|^2}$$

By the linearity of inner products

$$|\langle Q\mathbf{w}_0, Q\mathbf{x} \rangle - \langle \mathbf{w}_0, \mathbf{x} \rangle| = \frac{2}{\left\| \mathbf{x}_1^\star - \mathbf{x}_2^\star \right\|^2} |\left((\langle Q\mathbf{x}_1^\star, Q\mathbf{x} \rangle - \langle Q\mathbf{x}_2^\star, Q\mathbf{x} \rangle) - (\langle \mathbf{x}_1^\star, \mathbf{x} \rangle - \langle \mathbf{x}_2^\star, \mathbf{x} \rangle) \right)|$$

$$\leq \frac{6}{\left\| \mathbf{x}_1^\star - \mathbf{x}_2^\star \right\|^2} \chi' = 3 \left\| \mathbf{w}_0 \right\|^2 \chi'$$

We now have the necessary results to prove Prop. 13. By step. 1 we have the distance between the two supports is bounded by δ. By step. 8 we then have another solution $\mathbf{w}_0, b_0$ defined by the two supports. By step 9 we have the compressed inner product deviation property. Finally we show that $\|\mathbf{w}_0\| \leq \|\mathbf{w}^\star\|$. By the properties of closures and limits, Eqn 5.2 also holds for all $\mathbf{x}_1 \in \text{cl}(\text{co}(\mathcal{G}^+))$ and $\mathbf{x}_2 \in \text{cl}(\text{co}(\mathcal{G}^-))$. Thus, $\|\mathbf{w}^\star\| \geq \frac{2}{\|\mathbf{x}_1^\star - \mathbf{x}_2^\star\|} = \|\mathbf{w}_0\|$ which completes the proof.

Finally, we can prove theorem 17. By construction. m satsifies that the inner product deviates less than 1 so we can scale the compressed $\mathbf{w}_0$ and it satisfies conditions

6. MULTICLASS LEARNING WITH ERROR CORRECTING CODES

6.1 Problem Setup

So far we have restricted our attention to binary classification in this dissertation. However, many applications arise in which the label set is not a binary class but is a multiclass. For example, ImageNet contains images of various objects such cats, dogs, houses, etc and the algorithm is tasked with selecting the correct category for each image [8]. We note that the number of categories is not infinite ($|\mathcal{Y}| \leq \infty$). Regardless of what the label is, we assume $\mathcal{Y} = [k]$ for simplicity. That is, each of the k element labels is mapped to an integer.

While complex multiclass functions f exists, there is a certain practical and computational ease of binary classification. Additionally, many powerful binary classification algorithm already exists. This motivates reducing a multiclass classification task into a series of binary classification tasks. In ImageNet for example, one can imagine first classifying whether the image is a dog or something else. Then classifying whether the image is a cat or something else. And so on for each class. This approach is commonly called One-vs-all (OvA) and reduces the multiclass task into k binary classification tasks. Another approach would be to first classify whether the image is a dog or a cat. Then classify where the image is a dog or a house. And so on for each pair of classes. This approach is commonly called All-Pairs (AP) and reduces the multiclass task into $k(k-1)/2$ binary classification tasks. A more complex reduction proposed by Dietterich and Bakiri [9] is based on error correcting output codes (ECOC).

These approaches can all be defined in the uniform framework proposed by Allwein et al. [1] and Dietterich and Bakiri [9]. For each of these reductions from a k-class multiclass task to l binary classifications tasks, a coding matrix $\boldsymbol{M} \in \{-1, 0, 1\}^{k \times l}$ is defined. Each of the l binary classification tasks receive a training set where the multiclass label has been mapped based on $\boldsymbol{M}$ i.e. the i-th binary classifier receives the training set $((\mathbf{x}_j, \boldsymbol{M}(y_j, i)))_{j \in [m] \text{s.t.} \boldsymbol{M}(y_j,i) \in \{-1,1\}}$. To elaborate, the i-th binary classifier uses the i-th column of $\boldsymbol{M}$, $\boldsymbol{M}(., i)$, as the map. The y-th

element in $\boldsymbol{M}(.,i)$, $\boldsymbol{M}(y,i)$, is the map for the multiclass label y. A point $(\mathbf{x}_j, y_j)$, in the original training set is excluding from the i-th binary classifier if $\boldsymbol{M}(y_j,i) = 0$. Otherwise, the point is included in the i-th binary classifier's training set as $(\mathbf{x}_j, \boldsymbol{M}(y_j,i))$.

We will now go construct the coding matrix for the popular One-vs-All and All-Pairs framework for a $k = 4$ multiclass task. For one-vs-all, the coding matrix is

$$\boldsymbol{M} = \begin{bmatrix} 1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{bmatrix}$$

Note that the first column of $\boldsymbol{M}$ corresponds to a classifier which classifies the data as $y = 1$ against the rest of the classes. Similarly for the second, third and fourth columns of $\boldsymbol{M}$ and the classes $y = 2, 3, 4$ respectively. For all-pairs, the coding matrix is

$$\boldsymbol{M} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 & -1 & -1 \end{bmatrix}$$

Note that the first column of $\boldsymbol{M}$ corresponds to a classifier which classifies the data with class 1 versus class 2 with all training points with class $3, 4$ ignored. The second column of $\boldsymbol{M}$ is class 1 versus class 3 and so on.

With the binary classification algorithm, $\mathcal{A}_b$, based on a binary margin function class $\mathcal{F}_b$, and the coding matrix, we can precisely define the multiclass learning algorithm in Algorithm 1.

6.2 Decoding

After running our multiclass learning algorithm with the coding matrix $\boldsymbol{M}$, we have as output $\hat{\mathbf{h}} = \hat{h}_1, \ldots, \hat{h}_l$. We now must define a multiclass prediction based in $\hat{\mathbf{h}}$. The two common decoding

Algorithm 1 Multiclass Learning as Reduction to Binary Classification

Input: Binary Alg $\mathcal{A}_b$
for $i = 1$ **to** $m - 1$ **do**
  $\hat{h}_i = \mathcal{A}_b\left(((\mathbf{x}_j, \boldsymbol{M}(y_j, i)))_{j \in [m] \text{s.t.} \boldsymbol{M}(y_j,i) \in \{-1,1\}}\right)$
end for

methodologies are Hamming decoding and loss-based decoding. These are equivalent to hard-decision decoding and soft-decision decoding in the coding theory literature.

Hamming Decoding: Hamming decoding only considers the binary class, sign $\left(\hat{h}_i(x)\right)$, and not the confidence, $|\hat{h}_i(x)|$, of the individual predictions. That is, Hamming decoding considers only the binary output of the individual binary classifiers. Following the notation from Allwein et al. [1], we define the Hamming distance between our output vector $\hat{\mathbf{h}}$, domain point, $\mathbf{x}$, and the r-th row in our coding matrix $M(r)$ as

$$d_{\mathbf{H}}\left(\boldsymbol{M}(r), \hat{\mathbf{h}} \circ \mathbf{x}\right) = \sum_{s=1}^{l} \left(\frac{1 - \boldsymbol{M}(r,s)\text{sign}(\hat{h}_s(\mathbf{x}))}{2} \right).$$

The predicted multiclass label is the label that minimizes the Hamming distance. That is,

$$\hat{y} = \underset{y \in [k]}{\text{argmin}} \; d_{\mathbf{H}}\left(\boldsymbol{M}(r), \hat{\mathbf{h}} \circ \mathbf{x}\right). \tag{6.1}$$

Loss-based Decoding: Loss-based decoding incorporates the confidence in the individual predictors as well as the binary class to output a multiclass label. We define the correlation of the output functions and row r of the coding matrix as

$$C_L\left(\boldsymbol{M}(r), \hat{\mathbf{h}} \circ \mathbf{x}\right) = \sum_{s=1}^{l} \boldsymbol{M}(r,s)\hat{h}_s(\mathbf{x}). \tag{6.2}$$

The predicted multiclass label is the label that maximizes the correlation. That is,

$$\hat{y} = \underset{y \in [k]}{\text{argmax}} \; C_L\left(\boldsymbol{M}(r), \hat{\mathbf{h}} \circ \mathbf{x}\right). \tag{6.3}$$

6.3 Previous Work

In this paper, we focus on improving the multiclass classification algorithms by improving the design of the encoding matrix M.

Several previous attempts have been made to successfully design the coding matrix. The first attempt to design the encoding matrix was to restrict the attention to a specific family of codes. For example, the All-Pairs and One-vs-All approach explained above can both be used to generate a encoding matrix [9]. The family of random codes were explored by Allwein et al. [1] and Dietterich and Bakiri [9]. Using the tools from coding theory, Dietterich and Bakiri [9] experimented with the class of linear error correcting codes and Guruswami and Sahai [12] explored using Hadamard-matrix codes. Each of these families of codes can be considered a problem-independent approach as the coding matrix does not depend on the particular problem at hand. Results in Allwein et al. [1] suggest that using a problem-dependent code can yield higher prediction accuracies. This problem stems from the fact that a particular code may produce a partition that cannot be accurately solved by $\mathcal{A}_b$ [6].

Several problem-dependent methods have been considered as well. The first choice, to search for the best set of columns over every possible combination of columns, is generally considered intractable [5]. Another approach was to restrict the search space to coding matrices that can be represented as a binary tree [10]. This allows the ability to search in a greedy way. However, the structure of the binary tree forces the final code to be extremely sparse requiring a high number of binary classifiers.

The second problem-dependent approach to coding matrix design involves an approximation measure of the easiness of the partitions in the matrix. This similarity measure is denoted as F_b. In addition, F_r approximates the ability of M to distinguish between the multiclass labels. Zhao and Xing [22] used the two functions and some additional requirements to solve the following optimization problem

$$\max_{M} F_b(M) - \lambda_r F_r(M) - \lambda_c \sum_{i=1}^{l} ||M(i)||_2^2 \tag{6.4}$$

$$\text{s.t.} M \in \{-1, 0, 1\}^{k \times l} \tag{6.5}$$

$$\sum_{k=1}^{K} \mathbb{1}_{M(k,l)=1} \geq 1, \sum_{k=1}^{K} \mathbb{1}_{M(k,l)=-1} \geq 1 \ \forall l = 1, \ldots, L \tag{6.6}$$

$$\sum_{l=1}^{L} \mathbb{1}_{M(k,l)\neq 0} \geq 1 \ \forall k = 1, \ldots, K \tag{6.7}$$

where $F_b(M)$ is a measure of the separability of each binary partition of the coding matrix and $F_r(M)$ is a measure of the easiness of the partition. The last term is to ensure a sparse solution. The idea of this optimization problem is to create a matrix that allows each row to be decoded correctly while ensuring you avoid 'bad' partitions.

For approximating the easiness of the partitions, Zhao and Xing [22] assumed a similarity matrix $S \in \mathbb{R}^{k \times k}$. The matrix is assumed to be known beforehand. $S_{i,j}$ is a measure of similarity between the i-th and j-th classes. F_b, then, sums the similarity value of the pairs that are grouped together and subtracts the value of the pairs in opposite groups for each partition. Explicitly,

$$F_b(M) = \sum_{i=1}^{l} \sum_{j=1}^{k-1} \sum_{r=j}^{k} \left(2 \cdot \mathbb{1}_{[M(j,i)\neq M(r,i)]} - 1\right) S(i,j) \tag{6.8}$$

The function used to approximate the distinquishabilty of M by Zhao and Xing [22] is the average correlation of the rows of M. This can be precisely written as

$$F_r(M) = \sum_{i=1}^{k} \sum_{j=1}^{k} M(i)^\top M(j), \tag{6.9}$$

where $M(i)$ represents the i-th row of M

6.4 Optimized Encoder Algorithm

We propose a two-step process for designing the coding matrix $\boldsymbol{M}$. In the first step, after choosing the desired number of binary classifiers l, we construct a dense coding matrix $\boldsymbol{M}_{\text{code}}$ with the optimal minimum Hamming distance based on the results from channel coding. In our results, we use BCH codes to achieve the optimal minimum distance in the code. This is the same process from Dietterich and Bakiri [9], Guruswami and Sahai [12]. In the second step of our process, we optimize a permutation, $P : [k] \to [2^{k'}]$ injective, to achieve binary classifiers with high separability scores. This step can be thought of as reordering the rows of the coding matrix until the easiest binary partitions are found. The second step of the algorithm represents a new approach which is not found in Dietterich and Bakiri [9], Guruswami and Sahai [12].

6.4.1 Channel Code

In the first part of our algorithm we construct a dense coding matrix $\boldsymbol{M}_{\text{code}}$ based on the results from channel coding. Channel codes were developed to optimally transmit information across a noisy channel. We will focus on linear block codes which the sum of any two codewords is also a codeword. A (k', l', d_{min})-channel code is summarized by the number of input bits, k', the number of output bits, l', and the minimum distance between each codeword, d_{min}. A linear channel code is generally thought of as a generator matrix over the binary Galois field, $\boldsymbol{C} \in \text{GF}_2^{k' \times l'}$. The output message, o, sent across the channel is then the row vector input message, $m \in \text{GF}_2^{k'}$, multiplied by the generator matrix, $\boldsymbol{C}$. That is, $o = m\boldsymbol{C}$

To convert the channel code $\boldsymbol{C}$ into a suitable coding matrix for multiclass classification we need create matrix, $\boldsymbol{K}$, representing all labels as a k' length vector in the binary Galois field. We define the encoding of integer i into its k' length vector in the binary Galois field as $e : \mathbb{N} \to \text{GF}_2^{k'}$. Row i of $\boldsymbol{K}$ is then $\boldsymbol{K}_i = e(i)$. Then, the coding matrix, $\boldsymbol{M}_{\text{code}}$, is defined by multiplying the matrices $\boldsymbol{KC}$ and then mapping each elements of the binary Galois field to $\{-1, 1\}$. That is, the

map $g : \text{GF}_2 \to \{-1, 1\}$ maps 0 to -1 and 1 to 1.

$$\boldsymbol{M}_{\text{code}} = g(\boldsymbol{KC})$$

6.4.2 Optimal Permutation

6.4.3 Similarity Score:

To optimize the easiness of the binary partitions in a efficient way, we construct a similarity matrix (similar to Zhao and Xing [22]. The similarity matrix $\boldsymbol{S}$ is a $k \times k$ triangular matrix where the i, j entry represents how similar classes i and j are to each other. As we are working with extreme multiclass classification, we will need tho similarity matrix to be *sparse*. We require this as the storage complexity would be too great for these extreme multiclass problems. For example, a dense similarity matrix of the ODP dataset used later in this paper would be ~ 90 Gb.

The idea is that an easy binary partition with groups A and B will have the following properties: all classes in group A will have high similarity scores with each other, all classes in group B will have high similarity scores with each other, all pairs of classes $a \in A, b \in B$ from separate groups will have low similarity scores.

The similarity score of the matrix is defined [22]:

$$F_b(M) = \sum_{i=1}^{l} \sum_{j=1}^{k-1} \sum_{r=j}^{k} \left(2 \cdot \mathbb{1}_{\boldsymbol{M}_{j,i} \neq \boldsymbol{M}_{r,i}} - 1\right) \boldsymbol{S}_{i,j} \tag{6.10}$$

By using the fact that the Hamming distance between two rows p, q of the matrix can be written $d_{\mathbf{H}}(p, q) = \sum_{i=1}^{l} \mathbb{1}_{p(i) \neq q(i)}$ we can rewrite the similarity score for any general coding matrix M as

$$F_b(M) = \sum_{i=1}^{l} \sum_{j=1}^{k} \sum_{n=1}^{k} \left(\boldsymbol{S}_{j,n} \mathbb{1}_{M_{j,i} = M_{n,i}} - \boldsymbol{S}_{j,n} \mathbb{1}_{M_{j,i} \neq M_{n,i}}\right)$$

$$= \sum_{j=1}^{k} \sum_{n=1}^{k} \boldsymbol{S}_{j,n} \left(\sum_{i=1}^{l} \left(\mathbb{1}_{M_{j,i} = M_{n,i}} - \mathbb{1}_{M_{j,i} \neq M_{n,i}}\right) \right)$$

$$= \sum_{j=1}^{k} \sum_{n=1}^{k} \boldsymbol{S}_{j,n} \left(l - 2d_H(M_j, M_n)\right)$$

Then maximizing $F_b(M)$ is the same as minimizing the following

$$\sum_{j=1}^{k} \sum_{n=1}^{k} \boldsymbol{S}_{j,n} d_H(M_j, M_n)$$

Thus, the goal is to find a map $P : [k] \to [2^{k'}]$ injective that minimizes

$$\min \sum_{j=1}^{k} \sum_{n=1}^{k} \boldsymbol{S}_{j,n} d_H(M_{P(j)}, M_{P(n)})$$

We propose a fast greedy optimization procedure to this problem that utilizes the properties of the code to increase efficiency.

6.4.3.0.1 Algorithm: We exploit the fact that the similarity matrix is sparse to create a greedy algorithm that is extremely efficient. Our algorithm goes through the non-zero entries of the similarity matrix in order of descending absolute value. For each non-zero entry, if the value is positive, the algorithm assigns the row index and column index of the value two codewords that are minimum distance away. This means the two classes (the row index and the column index) are grouped in the same partition the maximum number of times allowed by the code chosen. If the non-zero entry was negative, the algorithm assigns the row index and the column index classes of the value to two codewords that are maximum distance away from each other. This ensures the fact that the two classes will be grouped in opposite classes the maximum number of times. We next explain how these codeword pairs can be efficiently computed with minimum overlap by exploiting the properties of the underlying channel code.

6.4.3.0.2 Codeword Pairs: For our algorithm, we need pairs of codewords that are minimum and maximum distance apart to efficiently assign codewords in a greedy fashion. We can exploit the properties of the code to create these pairs. Since the code is linear, the all-zero vector, $\vec{0}$, is a codeword. By looking at all codewords that are minimum distance and maximum distance from $\vec{0}$,

Algorithm 2 Greedy Optimize Mapping

Initialize set $usedIndices = \{\}$
Initialize set $usedCodes = \{\}$
Initialize $minCounter = 0$
Initialize $maxCounter = 0$
for (i, j) in sortperm($|S|$) **do**
  if sign($S_{i,j}$) = 1 **then**
    if $i \notin usedIndices$ and $j \notin usedIndices$ **then**
      while $\mathcal{C}(minCounter) \in usedCodes$ or $T_{\text{min}}(\mathcal{C}(minCounter)) \in usedCodes$ **do**
        $minCounter++$
      end while
      $P(i) = \mathcal{C}(minCounter)$
      $P(i) = T_{\text{min}}(\mathcal{C}(minCounter))$
      add i, j to $usedIndices$
      add $T_{\text{min}}(\mathcal{C}(minCounter)), \mathcal{C}(minCounter)$ to $usedCodes$
    end if
  else
    if $i \notin usedIndices$ and $j \notin usedIndices$ **then**
      while $\mathcal{C}(maxCounter) \in usedCodes$ or $T_{\text{max}}(\mathcal{C}(maxCounter)) \in usedCodes$ **do**
        $maxCounter++$
      end while
      $P(i) = \mathcal{C}(maxCounter)$
      $P(i) = T_{\text{max}}(\mathcal{C}(maxCounter))$
      add i, j to $usedIndices$
      add $T_{\text{max}}(\mathcal{C}(maxCounter)), \mathcal{C}(maxCounter)$ to $usedCodes$
    end if
  end if
end for
randomly add the rest

we then choose an index j such that there is a codeword c^1 minimum distance from $\vec{0}$ with $c^1_j = 1$ and a codeword c^2 maximum distance from $\vec{0}$ with $c^2_j = 1$.

Using this index, j, we can create a coset $\mathcal{C}$ of half all the possible codewords. This coset will have the property that every codeword, $c \in \mathcal{C}$, will have codeword minimum distance away, c_{min}, such that $c_{\text{min}} \notin \mathcal{C}$ and will have a codeword maximum distance away, c_{max}, such that $c_{\text{max}} \notin \mathcal{C}$.

Using this index, j, we define the coset as $\mathcal{C} = \{e(i)\boldsymbol{C} : i \in [2^{k'}] \text{ s.t. } (e(i)\boldsymbol{C})_{i,j} = \text{GF}_2(0)\}$. The properties of a coset can be easily verified from this definition as any two codewords in $\mathcal{C}$ added together will also have $GF_2(0)$ in index j.

The minimum distance pairs can then be easily created using c^1. That is, for codeword $c \in \mathcal{C}$, the minimum distance pair codeword is $c + c^1$. This creation has several benefits we can easily show: 1) $c + c^1$ is minimum distance away from c since c^1 is minimum distance away from $\vec{0}$ 2) Since $c^1_j = 1$ and $c_j = 0$ since $c \in \mathcal{C}$, $c + c^1 \notin \mathcal{C}$ 3) Any two distinct codewords $a, b \in \mathcal{C}$ will have a distinct minimum distance pair by the linearity of the code.

The maximum distance pairs can be create in the same way using codeword c^2 instead and will satisfy the same properties. The overall lack of overlap in the creation of the pairs will allow for increase performance in the greedy optimization procedure. Once we have the coset and the minimum and maximum codeword pairs, we are able to run the algorithm in 2

6.4.3.0.3 Final Coding Matrix: After we find our optimized permutation $P : [k] \to [2^{k'}]$ injective, we must create the final coding matrix. We create a matrix $\boldsymbol{P} \in \mathrm{GF}_2^{k \times k'}$ with each row having the binary representation of the mapped value for that row. That is,

$$\boldsymbol{P} = \begin{bmatrix} e(P(1)) \\ e(P(2)) \\ e(P(3)) \\ \vdots \\ e(P(k)) \end{bmatrix}$$

The output coding matrix is then

$$\boldsymbol{M} = g(\boldsymbol{PC})$$

6.4.4 Algorithm Analysis

Our proposed algorithm is extremely fast. The only step that scales asymptotically with k is the initial sorting of the values of $\boldsymbol{S}$. The other steps in the algorithm are constant time computational steps such as set containment and array indexing. As the similarity matrix is row-sparse, there are $\mathcal{O}(k)$ non-zero values in $\boldsymbol{S}$ and thus the sorting takes $\mathcal{O}(k \log k)$ time.

We briefly compare this to the algorithm presented in Zhao and Xing [22]. Their algorithm

requires running steps of taking $k \times k$ matrix multiplication and $k \times k$ matrix inversions until convergence is achieved. That is, the inner loop complexity of their algorithm is $\mathcal{O}(k^3)$ and that inner loop has to be run until convergence. That computational cost is much greater than the $\mathcal{O}(k \log k)$ total cost of our algorithm that comes from exploiting the properties of the code.

Additionally, as our algorithm stores the codeword pairs by the index number that generates the codeword, the total storage cost of these pairs is $\mathcal{O}(k)$. The cost combined with the storage cost of the similarity matrix, also $\mathcal{O}(k)$, means our algorithm also has an efficient storage cost. For comparison, the $k \times k$ matrix multiplication and $k \times k$ matrix inversions required by the optimization procedure in Zhao and Xing [22] requires RAM storage of $\mathcal{O}(k^2)$. The total size of a $k \times k$ matrix for the ODP dataset presented in this paper would be ~ 90Gb. This is too large for even a high end personal computer.

6.5 Theory

Our optimization procedure differs from previous attempts using the similarity matrix by forcing the coding matrix to achieve an optimum minimum Hamming distance. We require this optimality based on the following theoretical result regarding the generalization of multiclass learning.

6.5.1 Previous Work

Maximov and Reshetova [15] analyzed a generalized multiclass margin classifier setup that relied on separate function classes for each potential label. That is, there is a function class for label 1, $\mathcal{F}_1$, label 2, $\mathcal{F}_2$ etc. The algorithm then picks a vector of function $\vec{f} \in \mathcal{F}_1 \times \mathcal{F}_2 \ldots \mathcal{F}_k$. The output label for a point $\mathbf{x}$ is chosen by $\max_{i \in [k]} \vec{f}_i(\mathbf{x})$. They showed that the generalization bound of multiclass margin classification grows linearly with respect to the number of classes when looking at these function classes. They additionally showed that this growth rate is tight with respect to their assumptions. They achieved these results using by looking at the Rademacher complexities.

Applying their result to the our setup, we get the growth rate of the generizalion bound is kl with respect to the binary hypotheses class $\mathcal{H}_b$ where k is the number of classes and l is the number of columns in the coding matrix M. This result follows from noting that the label function class

$\mathcal{F}_y$ has a Rademacher complexity of $lR(\mathcal{H}_b \circ \mathcal{S})$. This can be easily shown by following the steps in lemma 20 and fixing the row of $\boldsymbol{M}$ instead of allowing it to change with the label y in z.

6.5.2 Hard Decoding

In this section, we present a generalization bound for hard decoding with a coding matrix. We note that this setup is more restrictive than the setup analyzed in Maximov and Reshetova [15]. This restriction is what allows us to beat their bound as it was shown the bound was tight. Our results show that if the coding matrix is chosen with an optimum hamming distance, the generalization bound is constant with respect to the number of classes. This is a substantial improvement over Maximov and Reshetova [15].

Since we are looking only in terms of hard decoding, we consider the binary hypothesis class $\mathcal{H}_b^{\boldsymbol{H}}$ that maps each function from $\mathcal{H}_b$ to a hard decision. That is, $\mathcal{H}_b^{\boldsymbol{H}} = \{x \mapsto \text{sign}(h(x)) \mid \forall h \in \mathcal{H}_b\}$. First, we define a multiclass function class $\hat{\mathcal{F}}$ for the coding matrix setup can be written as the following:

$$\hat{\mathcal{F}} = \left\{ z \mapsto \sum_{j \in [l]} \mathcal{M}(y, j) h_j(x) : \forall \vec{h} \in \mathcal{H}_b^{\boldsymbol{H} \otimes l} \right\} \tag{6.11}$$

Now, we show that bound the Rademacher complexity of the multiclass function class in terms of the binary function class.

Lemma 20. *The Rademacher complexity of the set of function, $\hat{\mathcal{F}}$, defined in eqn. 6.11 is*

$$R(\hat{\mathcal{F}}) = lR(\mathcal{H}_b^{\boldsymbol{H}})$$

Proof:

$$
\begin{aligned}
mR(\hat{\mathcal{F}}) &= \mathbb{E}_\sigma \sup_{a \in \hat{\mathcal{F}} \circ \mathcal{S}} \sum_{i \in [m]} \sigma_i a_i \\
&= \mathbb{E}_\sigma \sup_{f \in \hat{\mathcal{F}}} \sum_{i \in [m]} \sigma_i f(x_i, y_i) \\
&= \mathbb{E}_\sigma \sup_{\vec{h} \in \mathcal{H}_b^{\boldsymbol{H}}} \sum_{i \in [m]} \sigma_i \sum_{j \in [l]} \mathcal{M}(y_i, j) h_j(x_i) \\
&= \mathbb{E}_\sigma \sup_{\vec{h} \in \mathcal{H}_b^{\boldsymbol{H}}} \sum_{j \in [l]} \sum_{i \in [m]} \sigma_i \mathcal{M}(y_i, j) h_j(x_i) \\
&= \mathbb{E}_\sigma \sum_{j \in [l]} \sup_{h_j \in \mathcal{H}_b^{\boldsymbol{H}}} \sum_{i \in [m]} \sigma_i \mathcal{M}(y_i, j) h_j(x_i) \\
&= \sum_{j \in [l]} \mathbb{E}_\sigma \sup_{h_j \in \mathcal{H}_b^{\boldsymbol{H}}} \sum_{i \in [m]} \sigma_i \mathcal{M}(y_i, j) h_j(x_i)
\end{aligned}
$$

As you are just flipping the equal probability σ_i

$$
\begin{aligned}
mR(\hat{\mathcal{F}}) &= \sum_{j \in [l]} \mathbb{E}_\sigma \sup_{h_j \in \mathcal{H}_b^{\boldsymbol{H}}} \sum_{i \in [m]} \sigma_i h_j(x_i) \\
&= \sum_{j \in [l]} mR(\mathcal{H}_b^{\boldsymbol{H}}) \\
&= lmR(\mathcal{H}_b^{\boldsymbol{H}})
\end{aligned}
$$

We then define a multiclass margin based on the following selected parameter $u \in (l - \frac{d}{2}, l]$. Let the margin function be the following

$$
M(x) := \begin{cases} 1 & x < l - d \\ \frac{x}{l-d-u} - \frac{u}{l-d-u} & l - d \leq x \leq u \\ 0 & x > u \end{cases}
$$

and let $\mathcal{M} = M \circ \hat{\mathcal{F}}$. Then for all $m \in \mathcal{M}$

$$\mathbb{P}_{z\sim\mathcal{D}}\left[\text{ECOC}(\hat{h}\circ x) \neq y\right]$$

$$= \mathbb{E}_{z\sim\mathcal{D}}\left[\mathbb{1}_{\text{ECOC}(\hat{h}\circ x)\neq y}\right]$$

$$= \mathbb{E}_{z\sim\mathcal{D}}\left[\mathbb{1}_{\hat{f}(x)\leq l-d/2}\right]$$

$$\leq \mathbb{E}_{z\sim\mathcal{D}}\left[m(x)\right]$$

and

$$\mathbb{P}_{z\sim\mathcal{S}}\left[\hat{f}(z) \geq u\right] = \mathbb{E}_{z\sim\mathcal{S}}\left[\mathbb{1}_{\hat{f}(x)\geq u}\right] \geq \mathbb{E}_{z\sim\mathcal{S}}[m(x)]$$

Additionally, the Rademacher complexity of $\mathcal{M}$ is bounded by the following by the contraction lemma for Rademacher complexities [18] and the fact that $M(x)$ is Lipschitz.

$$R(\mathcal{M}) \leq \frac{1}{d+u-l}R(\hat{F}) \leq \frac{l}{d+u-l}R(\mathcal{H}_b^{\boldsymbol{H}})$$

Based on these results and the fact that $\forall m \in \mathcal{M}\ |m(z)| \leq 1$, we can apply Thm. 26.5 from Shalev-Shwartz and Ben-David [18] and we get

$$\mathbb{E}_{z\sim\mathcal{D}}\left[m(x)\right] \leq \mathbb{E}_{z\sim\mathcal{S}}\left[m(x)\right] + 2R(\mathcal{M}\circ\mathcal{S}) + 4\sqrt{\frac{2\ln(4/\delta)}{m}}$$

$$\mathbb{P}_{z\sim\mathcal{D}}\left[\text{ECOC}(\hat{h}\circ x) \neq y\right] \leq \mathbb{P}_{z\sim\mathcal{S}}\left[\hat{f}(z) \geq u\right] + \frac{l}{d+u-l}R(\mathcal{H}_b^{\boldsymbol{H}}) + 4\sqrt{\frac{2\ln(4/\delta)}{m}}$$

6.5.2.1 Discussion

The intuition for the bound comes from the properties of hard decoding. A hard decoding algorithm is guaranteed to be correct if the number of errors is less than the half of the minimum

distance of the coding matrix. The values of the output of any $f \in \hat{\mathcal{F}}$ are between $-l$ and l. The semantic meaning of the output of f is the number of right positions minus the number of positions that are wrong. This makes the number of errors $\frac{l-f(x)}{2}$. Thus, if the value of f is greater than $l-d$, the number of errors is less than $d/2$ and the hard decoding is guaranteed to be correct.

We will now discuss some insights the bound can give us for several coding matrix designs. First, if we look at a OvA coding matrix, we have that $l = k$, $d = 2$, and $u = \mathcal{O}(k)$. This gives a $\mathcal{O}(k)$ growth rate for the second term. We note that this is an improvement over the tight kl growth rate in Maximov and Reshetova [15]. We are able to beat the growth rate by looking at a more restrictive problem definition with a coding matrix and using hard decoding.

We are able to beat the growth rate even more if we use an code with an optimal minimum Hamming distance. If we choose a simplex code with a length $l = k$, then the simplex code gives a matrix with minimum distance $d = \lfloor k/2 \rfloor$. This means $u = O(k)$ which makes the second term constant and independent of k. This means that the decoding accuracy bound does not grow with any explicit dependence on k if we use a coding matrix based on a simplex code.

6.5.3 Soft Decoding

In this section, we present a generalization bound for soft decoding with a coding matrix. For this analysis, we require that the codewords of the dense coding matrix be a subset of a linear code.

We define the vector $\vec{h} \in \mathcal{H}_b^{\otimes l}$ which is thought of as the learning algorithm output vector of binary functions. We analyze the class of margin classifiers $\mathcal{M}$ where $\forall m \in \mathcal{M}$ we have $m : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$

$$\mathcal{M} = \left\{ \langle \boldsymbol{M}_y, \vec{h}(x) \rangle - \max_{y' \neq y} \langle \boldsymbol{M}_{y'}, \vec{h}(x) \rangle \mid \forall \vec{h} \in \mathcal{H}_b^{\otimes l} \right\}$$

We note that a margin classifier $m \in \mathcal{M}$ classifies a point $(\mathbf{x}, y)$ correctly if $m(\mathbf{x}, y) \geq 0$.

Now, we can compute the Rademacher complexity of the class of margin classifiers as

$$mR(\mathcal{M}) = \mathbb{E}_\sigma \sup_{m \in \mathcal{M}} \sum_{i \in [m]} \sigma_i m(\mathbf{x}_i, y_i)$$

$$= \mathbb{E}_\sigma \sup_{\vec{h} \in \mathbf{H}_b^{\otimes l}} \sum_{i \in [m]} \sigma_i (\langle \boldsymbol{M}_{y_i}, \vec{h}(x_i) \rangle - \max_{y' \neq y_i} \langle \boldsymbol{M}_{y'}, \vec{h}(x_i) \rangle)$$

$$= \mathbb{E}_\sigma \sup_{\vec{h} \in \mathbf{H}_b^{\otimes l}} \sum_{i \in [m]} \sigma_i \left(\max_{y' \neq y_i} \left(\sum_{j \in [l]} (\boldsymbol{M}_{y_i,j} - \boldsymbol{M}_{y',j}) \, \vec{h}_j(x_i) \right) \right)$$

By the fact that all $\boldsymbol{M}_{i,j} \in \{1, -1\}$,

$$= \mathbb{E}_\sigma \sup_{\vec{h} \in \mathbf{H}_b^{\otimes l}} \sum_{i \in [m]} \sigma_i \left(\max_{y' \neq y_i} \left(\sum_{j \text{ s.t. } \boldsymbol{M}_{y_i,j} \neq \boldsymbol{M}_{y',j}} 2 \boldsymbol{M}_{y_i,j} \vec{h}_j(x_i) \right) \right)$$

Since all rows of $\boldsymbol{M}$ are from a linear code, the maximum number of positions the rows can be different is $l - d$ where d is the minimum distance of the coding matrix. This gives

$$\leq \mathbb{E}_\sigma \sup_{\vec{h} \in \mathbf{H}_b^{\otimes l}} \sum_{i \in [m]} \sigma_i 2(l-d) \max_{j \in [l]} \left| \vec{h}_j(x_i) \right|$$

$$\leq 2(l-d) \mathbb{E}_\sigma \sup_{\vec{h} \in \mathbf{H}_b^{\otimes l}} \sum_{i \in [m]} \sigma_i \sum_{j \in [l]} \left| \vec{h}_j(x_i) \right|$$

$$= 2(l-d) \mathbb{E}_\sigma \sum_{j \in [l]} \sup_{\vec{h}_j \in \mathcal{H}} \sum_{i \in [m]} \sigma_i \left| \vec{h}_j(x_i) \right|$$

$$= 2(l-d) \sum_{j \in [l]} \mathbb{E}_\sigma \sup_{\vec{h}_j \in \mathcal{H}} \sum_{i \in [m]} \sigma_i \left| \vec{h}_j(x_i) \right|$$

By the contraction lemma of Lipschitz functions for Rademacher complexities and the fact $|.|$ is 1-Lipschitz

$$\leq 2(l-d) \sum_{j \in [l]} \mathbb{E}_\sigma \sup_{\vec{h}_j \in \mathcal{H}} \sum_{i \in [m]} \sigma_i \vec{h}_j(x_i)$$

$$= 2(l-d) \sum_{j \in [l]} m R(\mathcal{H}_b) = 2(l-d) l m R(\mathcal{H}_b)$$

Now, we can define the training margin loss as $\mathbb{E}_{z \sim \mathcal{S}}[m(x) \leq \theta]$ and use the ramp loss seen in many Rademacher bounds to bound the multiclass true loss by

$$\mathbb{E}_{z \sim \mathcal{D}}[m(x) \leq 0] \leq \mathbb{E}_{z \sim \mathcal{S}}[m(x) \leq \theta] + \frac{2}{\theta} l(l-d) R(\mathcal{H}_b) + 4 \sqrt{\frac{2 \ln(4/\delta)}{m}}$$

6.5.3.1 Discussion

We note two points about this bound. The first insight is motivated by a coding matrix designed by OvA with $l = k$ and $d = 2$. The only term in the bound that explicitly grows with k is the second term. For the OvA case, this term grows $\mathcal{O}(k^2)$. This is the same growth rate as Maximov and Reshetova [15]. By forcing the structure of a coding matrix yet designing the coding matrix using OvA, we do not get any improvement in the bound. Additionally, by using a simplex code with $l = k$ and $d = k/2$, we get the same asymptotic growth. However, while the order of the growth is the same, it has been cut in half by the minimum Hamming distance

The second insight is motivated by a coding matrix designed using an Expander code. This coding matrix would have $l = \mathcal{O}(\log k)$ and $d = \mathcal{O}(\log k)$. Thus, the only term in the bound that explicitly depends on k grows $\mathcal{O}((\log k)^2)$. By forcing the structure of a coding matrix and using a efficient coding matrix we get a major improvement from the tight $\mathcal{O}(k \log k)$ bound[15].

There are also two consideration we must take when discussing these bounds. First, the asymptotic analysis assumes that the training margin loss does not grow with k. While this cannot be formally proven, it is a general assumption used when discussing these bounds [15]. Second, the better asymptotic growth rate of using an efficient code in place of OvA does not guarantee that the code will perform better. It may be impossible to achieve a small margin loss using the efficient coding matrix making the bound vacuous even though it grows slowly. Whereas the OvA coding matrix may have a very small margin loss making the bound non-vacuous even with the worse growth rate.

Despite these considerations, we believe the bounds motivate using a coding matrix with an optimum Hamming distance. However, we must ensure that the binary partitions are "easy" so we can keep the training margin loss low. We present our empirical results in the next section showing our algorithm can accomplish this.

6.6 Results

We will show in our empirical results that: 1) our greedy algorithm works successfully to produced a coding matrix with a high similarity score 2) Learning with our optimized coding matrix produced an ECOC algorithm with high testing accuracy. We will show these results on the following four datasets:

Table 6.1: Datasets

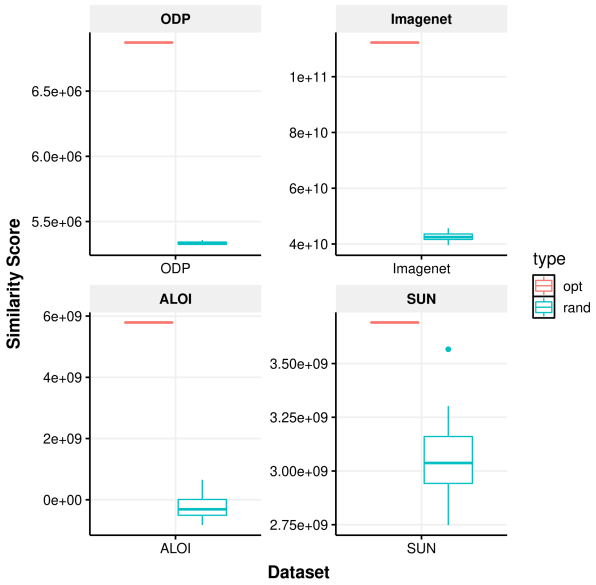
Dataset	Class Size	Feature Size	#Train/#Test
ODP	105033	493014	1084404 / 493014
ImageNet	21841	6144	12777062 / 1419674
ALOI	1000	82944	86400 / 9600
SUN	397	90000	97879 / 10875

6.6.1 Similarity Optimization

First, we show that our greedy optimization procedure works for optimizing the similarity score. As our optimization procedure is greedy, it is not guaranteed to be optimal. However, we show empirically that it is much better than is any random permutation.

In figure 6.1, we show the range of matrix similarity scores computed with random permutations as well as the similarity score of our optimized permutation for each dataset. As seen in the figure, the optimized permutation perform much better than all of the random permutations.

Figure 6.1: ODP Similarity Score



We summarize the benefits of the optimization procedure in tables 6.2 and 6.3 as well. In table 6.2 we show optimized permutation similarity scores and the average similarity score of a random permutation and well as the standard deviation of those scores. In table 6.3, we show the number of standard deviations better the optimized permutation when compared with the random permutation mean.

Table 6.2: Similarity Score

Dataset	Random Permutation		Optimized
	Mean	Std Dev	
ODP	5.34×10^6	1.30×10^4	6.87×10^6
Imagenet	4.26×10^{10}	1.62×10^9	1.12×10^{11}
ALOI	-2.53×10^8	3.78×10^8	5.79×10^9
SUN	3.06×10^9	1.86×10^8	3.69×10^9

Table 6.3: Similarity Score Better

Dataset	# Stddev Better
ODP	117.69
Imagenet	42.88
ALOI	15.97
SUN	3.38

Table 6.4: Number of Binary Classifiers

Dataset	# Classifiers
ODP	1014
Imagenet	1022
ALOI	63
SUN	122

6.6.2 Testing Accuracy

We now present how well our procedure works on real datasets. As we are working on extreme multiclass problems, the OvA and AP approaches are not computationally feasible. Even with a parallelized algorithm on a 8 core computer, computing each binary classifier and decoding each example would take over a year with our implementation. We instead compare our algorithm a random matrix, shown by Allwein et al. [1] and Rifkin and Klautau [17], to be comparable to OvA. Additionally, we compare our algorithm to a coding matrix generate with a random permutation to see the benefit of optimizing the permutation with respect to the similarity matrix.

We use a BCH code for each of the code-based coding matrices. We use the number of binary classifier listed in table 6.4 for each dataset. We use vowpalwabbit [] to compute the binary classifiers.

The accuracies presented are whether the real label is in the top prediction (Top 1), in the top 5 predictions (Top 5), or in the Top 10 predictions (Top 10). We plot the decoding accuracy for both hard and soft decoding for the ODP dataset, figure 6.2, the Imagenet dataset, figure 6.3, the ALOI dataset, figure 6.4, and the SUN dataset, figure 6.5.

Figure 6.2: ODP Decoding Accuracy

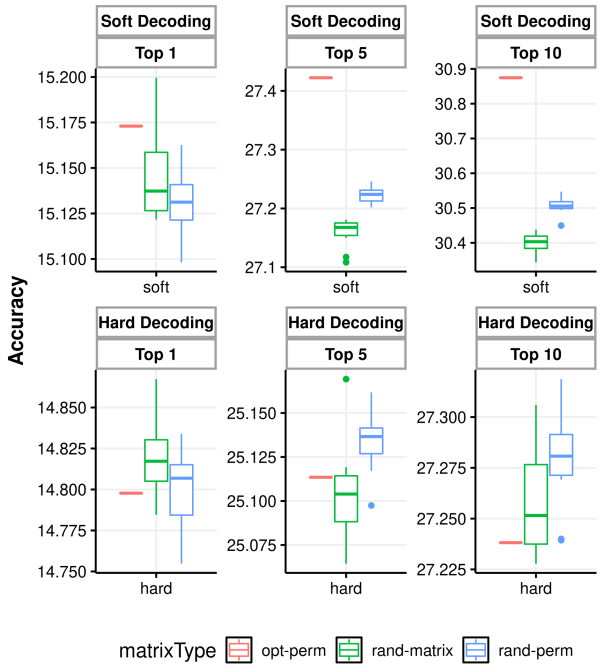


Figure 6.3: Imagenet Decoding Accuracy

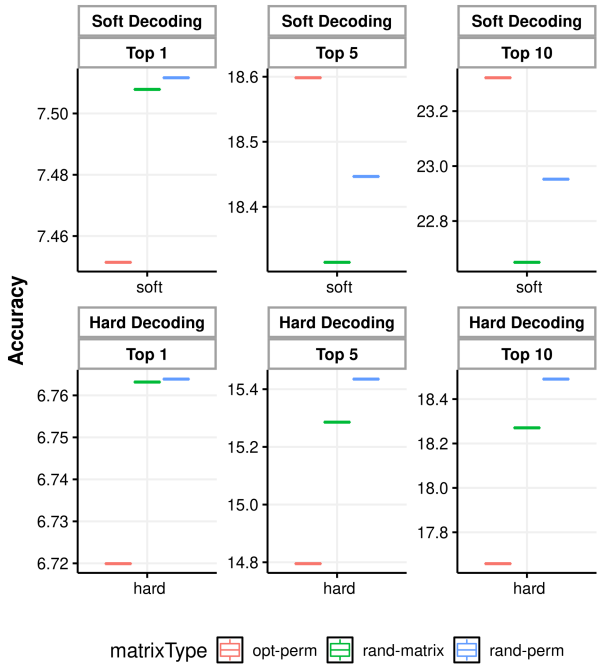


Figure 6.4: ALOI Decoding Accuracy

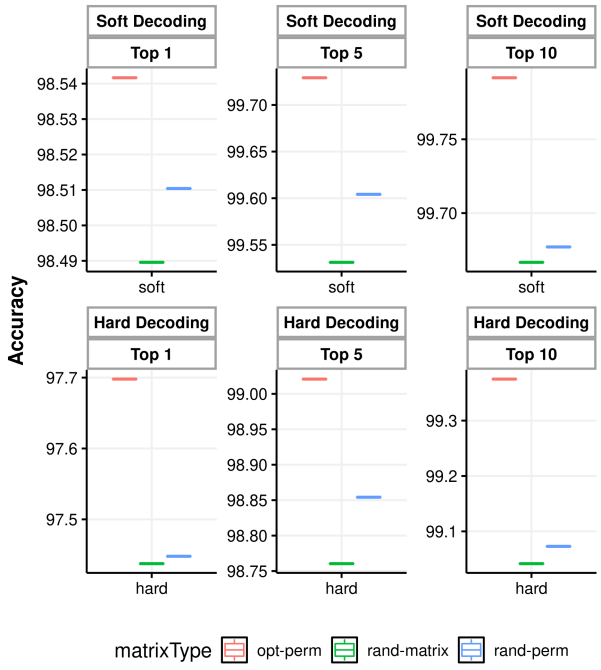


Figure 6.5: SUN Decoding Accuracy

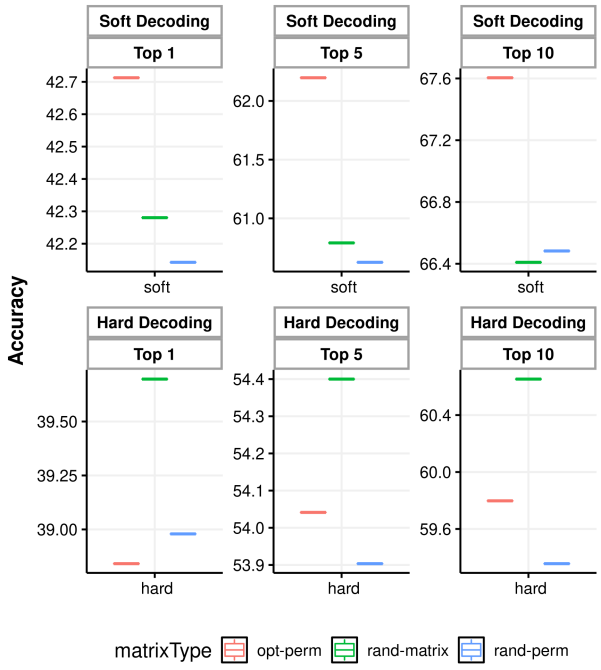


Table 6.5: Soft Decoding Accuracies (in Percent)

Coding Matrix	ODP			ImageNet		
	Top 1	Top 5	Top 10	Top 1	Top 5	Top 10
Random Matrix	15.14	27.16	30.40	7.51	18.31	22.65
Random Permutation	15.13	27.22	30.51	**7.51**	18.45	22.95
Opt Permutation	**15.17**	**27.42**	**30.87**	7.45	**18.60**	**23.32**
Coding Matrix	ALOI			SUN		
	Top 1	Top 5	Top 10	Top 1	Top 5	Top 10
Random Matrix	98.49	99.53	99.67	42.28	60.79	66.41
Random Permutation	98.51	99.60	99.68	42.14	60.63	66.48
Opt Permutation	**98.54**	**99.73**	**99.79**	**42.71**	**62.20**	**67.60**

Table 6.6: Hard Decoding Accuracies (in Percent)

Coding Matrix	ODP			ImageNet		
	Top 1	Top 5	Top 10	Top 1	Top 5	Top 10
Random Matrix	**14.82**	25.10	27.26	6.76	15.29	18.27
Random Permutation	14.80	**25.13**	**27.28**	**6.76**	**15.43**	**18.49**
Opt Permutation	14.79	25.11	27.24	6.72	14.80	17.66
Coding Matrix	ALOI			SUN		
	Top 1	Top 5	Top 10	Top 1	Top 5	Top 10
Random Matrix	97.44	98.76	99.04	**39.69**	**54.40**	**60.65**
Random Permutation	97.45	98.85	99.07	38.98	53.90	59.36
Opt Permutation	**97.70**	**99.02**	**99.38**	38.84	54.04	59.80

Additionally, we present the soft decoding accuracies in table 6.5 and the hard decode accuracies in table 6.6.

6.6.3 Discussion

There is a lot to unpack about these results presented in the plots and the tables.

When using soft or loss-based decoding, we can make several conclusions. First, it appears that the minimum makes at least a small difference when comparing the a random code-based coding matrix and the random matrix. This gives some credence to the appearance of the minimum Hamming distance in the generalization bound. However, the Rademacher complexity term is not the only term in the bound. We also have to worry about the training margin loss. The fact that the benefit is quite small reinforces the ideas presented in Rifkin and Klautau [17] that a code-based

coding matrix requires "weird" binary partitions that are hard for the binary learner to learn.

The optimized permutation coding matrix is the clear leader for all of the datasets. We show we are able to get the benefit of the minimum Hamming distance as motivated by our generalization bound while optimizing the coding matrix to make each of the binary partitions easy to learn.

The most interesting observation from the soft decision decoding is how the benefit increases substantially as you begin to look at the Top 5 and the Top 10 accuracies. As the random code-based coding matrix does not show this improvement, we must assume that the benefit is coming for creating easier partitions for the binary learner.

When looking at hard decoding, we cannot make any discernible observation. Each of the different matrix construction perform best in all accuracy measures on at least one of the datasets. This means the minimum hamming distance of the coding matrix does not play a large role in determining the output accuracy. Using a error correcting code and a random permutation (essentially a random encoder for the code) give about the same performance as a random matrix which obviously does not have an optimum minimum hamming distance.

However, this does completely invalidate the theoretical bound presented. As mentioned in the discussion in that section, there are three terms in the bound and we cannot know which term in dominating. It may be that the training hard decoding error is the dominating term for the true hard decoding error.

7. CONCLUSIONS

In this dissertation, we have shown that compressed learning for hard-SVM is possible. If we assume sparsity, we showed that if the restricted isometry constant of the compression matrix is bounded by $\frac{1}{\|\mathbf{w}^\star\|R^2}$ the separability assumption holds. If we cannot assume sparsity, we have shown that the separability assumption holds for a subgauassian random matrix if the compression length m satisfies $3\|\mathbf{w}^\star\|^2\left(4R(\frac{CK^2w(\mathcal{X})}{\sqrt{m}}) + (\frac{CK^2w(\mathcal{X})}{\sqrt{m}})^2\right) < 1$. Additionally, we showed after compression, the generalization bounds increases as $\frac{\mathcal{L}_B}{\sqrt{1-\delta_{2s}R^2\|\mathbf{w}^\star\|^2}}$.

We have presented and analyzed a new algorithm for selecting a error correcting output coding matrix for multiclass classification. This algorithm is extremely efficienct running in $\mathcal{O}(k\log k)$ time. We showed theoretically that the minimum Hamming distance is important for the generalization bound of multiclass classification and that our algorithm ensures a high minimum Hamming distance. Finally, we demonstrated on several extreme multiclass problems that our algorithm outperforms previous methods when using soft-decoding.

REFERENCES

[1] Erin L. Allwein, Robert E. Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *J. Mach. Learn. Res.*, 1:113–141, September 2001. ISSN 1532-4435. doi: 10.1162/15324430152733133. URL http://dx.doi.org/10.1162/15324430152733133.

[2] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[3] Kristin P Bennett and Erin J Bredensteiner. Duality and geometry in svm classifiers. In *ICML*, volume 2000, pages 57–64, 2000.

[4] Robert Calderbank, Sina Jafarpour, and Robert Schapire. Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain. 2009.

[5] Koby Crammer and Yoram Singer. On the learnability and design of output codes for multiclass problems. *Machine Learning*, 47(2):201–233, 2002. ISSN 1573-0565. doi: 10.1023/A:1013637720281. URL http://dx.doi.org/10.1023/A:1013637720281.

[6] Amit Daniely, Sivan Sabato, and Shai Shalev-Shwartz. Multiclass learning approaches: A theoretical comparison with implications. *CoRR*, abs/1205.6432, 2012. URL http://arxiv.org/abs/1205.6432.

[7] Hal Daumé III, Nikos Karampatziakis, John Langford, and Paul Mineiro. Logarithmic time one-against-some. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 923–932. JMLR. org, 2017.

[8] J. Deng, W. Dong, R. Socher, L. J. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009. doi: 10.1109/CVPR.2009.5206848.

[9] Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *CoRR*, cs.AI/9501101, 1995. URL http://arxiv.org/abs/cs.AI/9501101.

[10] Mongkon Eiadon, Luepol Pipanmaekaporn, and Suwatchai Kamonsantiroj. Mining discriminative class codes for multi-class classification based on minimizing generalization errors, 2016. URL http://dx.doi.org/10.1117/12.2242257.

[11] Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. Springer Science & Business Media, 2013.

[12] Venkatesan Guruswami and Amit Sahai. Multiclass learning, boosting, and error-correcting codes. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, COLT '99, pages 145–155, New York, NY, USA, 1999. ACM. ISBN 1-58113-167-4. doi: 10.1145/307400.307429. URL http://doi.acm.org/10.1145/307400.307429.

[13] Sham M Kakade, Karthik Sridharan, and Ambuj Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *Advances in neural information processing systems*, pages 793–800, 2009.

[14] Vladimir Koltchinskii and Dmitriy Panchenko. Rademacher processes and bounding the risk of function learning. In *High dimensional probability II*, pages 443–457. Springer, 2000.

[15] Yu Maximov and Daria Reshetova. Tight risk bounds for multi-class margin classifiers. *Pattern Recognition and Image Analysis*, 26(4):673–680, 2016.

[16] Milad Nasr, Amir Houmansadr, and Arya Mazumdar. Compressive traffic analysis: A new paradigm for scalable traffic analysis. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 2053–2069. ACM, 2017.

[17] Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *Journal of machine learning research*, 5(Jan):101–141, 2004.

[18] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014. ISBN 9781139952743. URL https://books.google.com/books?id=Hf6QAwAAQBAJ.

[19] Vladimir N Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity*, pages 11–30. Springer, 2015.

[20] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[21] Kilian Weinberger, Anirban Dasgupta, Josh Attenberg, John Langford, and Alex Smola. Feature hashing for large scale multitask learning. *arXiv preprint arXiv:0902.2206*, 2009.

[22] B. Zhao and E. P. Xing. Sparse output coding for large-scale visual recognition. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3350–3357, June 2013. doi: 10.1109/CVPR.2013.430.