# Ensemble Learning of High Dimension Datasets

A thesis

submitted in fulfilment

of the requirements for the Degree

of

Doctor of Philosophy in Statistics

at

The University of Waikato

by

**J.S. Lim**

KO TE TANGATA

THE UNIVERSITY OF
WAIKATO
*Te Whare Wānanga o Waikato*

**2020**

# Dedication

*To the memory of my grandmother.*

# Acknowledgements

I want to take this opportunity to express my appreciation and thanks to my supervisory panel with extra-special mention to my advisor Dr Robert J Durrant, for his support and guidance throughout this project. I would like to thank him especially for his extra time, wisdom, patience, advice and in keeping me focused on this project. His knowledge, insight, and encouragement have indeed been instrumental in helping me develop the necessary skills to complete this dissertation.

I would like to thank the Faculty and the School of Mathematics and Statistics at the University of Waikato for the opportunity to pursue my post-graduate studies, and the financial support awarded by the university. I appreciate especially the knowledge imparted by the professors and the way the special care from academic staff to graduate students. I would like to acknowledge my fellow post-graduates in the School of Mathematics and Statistics who have made the office entertaining and homely with the lively discussions and by mutually motivating and encouraging each other in our own projects.

I would like to thank Prof. Zhi-Hua Zhou, and Dr Ninh Dang Pham for examining and providing their valuable feedback on my thesis. I especially appreciate Prof Zhou, and Dr Pham for taking time away from their very busy schedule in order to do so.

I thank my friends in Hamilton, with special emphasis to my church group. They have been my family away from home, supporting me and keeping me motivated. I thank them especially for going the extra mile to make my post-graduate life a joy. I also would like to thank my friends back in Malaysia for their encouragement throughout my life as a post-graduate.

Last but absolutely not least, I would also like to thank my family members who have supported me and kept me in their daily thoughts and prayers. Their constant encouragement and support have been a driving force in completing this dissertation.

I dedicate this dissertation to my grandmother. I miss you Ah-Ma.

*Nick Lim Jin Sean*

# Abstract

Ensemble learning, an approach in Machine Learning, makes decisions based on the collective decision of a committee of learners to solve complex tasks with minimal human intervention. Advances in computing technology have enabled researchers build datasets with the number of features in the order of thousands and enabled building more accurate predictive models. Unfortunately, high dimensional datasets are especially challenging for machine learning due to the phenomenon dubbed as the “curse of dimensionality”. One approach to overcoming this challenge is ensemble learning using Random Subspace (RS) method, which has been shown to perform very well empirically however with few theoretical explanations to said effectiveness for classification tasks.

In this thesis, we aim to provide theoretical insights into RS ensemble classifiers to give a more in-depth understanding of the theoretical foundations of other ensemble classifiers. We investigate the conditions for norm-preservations in RS projections. Insights into this provide us with the theoretical basis for RS in algorithms that are based on the geometry of the data (i.e. clustering, nearest-neighbour). We then investigate the guarantees for the dot products of two random vectors after RS projection. This guarantee is useful to capture the geometric structure of a classification problem. We will then investigate the accuracy of a majority vote ensemble using a generalized Polya-Urn model, and how the parameters of the model are derived from diversity measures. We will discuss the practical implications of the model, explore the noise tolerance of ensembles, and give a plausible explanation for the effectiveness of ensembles.

We will provide empirical corroboration for our main results with both synthetic and real-world high-dimensional data. We will also discuss the implications of our theory on other applications (i.e. compressive sensing). Based on our results, we will propose a method of building ensembles for Deep Neural Network image classifications using RS projections without needing to retrain the neural network, which showed improved accuracy and very good robustness to adversarial examples. Ultimately, we hope that the insights gained in this thesis would make in-roads towards the answer to a key open question for ensemble classifiers, "When will an ensemble of weak learners outperform a single carefully tuned learner?"

*Keywords: Ensemble Learning, High Dimensional Datasets, Classification, Random Subspace Method*

# Contents

# List of Figures

# List of Tables

# 1

# Introduction

## 1.1 What is machine learning?

Machine Learning is the study of computer algorithms that learn a function from a series of examples without the function being explicitly programmed (Michie et al., 1994). Rather than engineering an algorithm that solves a complex task, the goal of machine learning is to "learn" the solution from provided examples and ultimately develop learning algorithms that solve complex tasks with minimal human intervention or assistance (Bengio et al., 2009).

The learning process requires information or data such that the algorithm may find patterns in the data to infer the decisions for the task based on the similarity to previous examples. This process may be based on a single sample, from which the computer returns an appropriate model decision function, or it may evolve as new data becomes available to the learning algorithm. Here we focus on the former setting, which is sometimes also called "statistical learning" (Vapnik, 1999). Machine learning tasks are usually categorized as follows based on the information available to the algorithms:

- Supervised: Target function outputs are available to the algorithm. Examples of supervised learning tasks include classification (labels are discrete values representing a category or class) and regression (labels are continuous values). Typical approaches include support vector machines, neural-networks, and least-squares regression (Joachims, 1998; MacKay, 2003).

- Unsupervised: Target labels are unavailable to the algorithm. Examples of unsupervised learning tasks include clustering, density estimation and outlier detection. Typical approaches include k-means, Gaussian mixture models, t-distributed stochastic neighbour embedding, and principal component analysis. (Ghahramani, 2003; Barber, 2012)

- Semi-supervised: Target labels are available for some data points with other data points, usually the majority, unlabelled. An example of a semi-supervised learning task is learning generative models to classify unlabelled data points (Zhu, 2005). Typical approaches are generative adversarial networks, co-training, and k-nearest neighbour graph (Goodfellow et al., 2014a; Zhou, 2017; Wang and Zhou, 2017).

- Reinforcement: Reinforcement learning aims to maximise the total reward through a series of decisions (called a policy). Examples of reinforcement learning include artificial intelligence system to play chess or to navigate a maze. Typical approaches are Q-learning, deep neural networks, one and multi-armed bandits (Sutton et al., 1998; Sutton, 1996; Wang et al., 2015).

### 1.1.1 Ensemble Learning

One popular “meta-learning” approach to machine learning is a technique known as “ensemble learning”. In ensemble learning, rather than having the decision made by a single learner e.g. one classifier, we train multiple learners and then combine the collective decisions into a single decision, for example by voting or averaging. The similarity of ensemble learning to the political systems in our world is not a coincidence. Ensemble learning takes inspiration from social and political science that the “wisdom of crowds” (a term borrowed from the Ancient Greeks) is *often* superior to the wisdom of “individuals”.

This confidence in the “wisdom of crowds” is not without merit. One commonly cited example supporting the “wisdom of crowds” is that at a 1906 country fair in Plymouth, when a crowd of 800 was asked to guess the weight of a slaughtered cow, the median guess of 1207 pounds was within 1% of the actual weight of the cow (1198 pounds) (Galton, 1907).

It is believed that the strength from the "wisdom of crowds" come from the diversity of opinions and the (approximate) independence of the individual members (Oinas-Kukkonen, 2008). In a diverse, independent group, each decision maker adds new information to the group decision and the group as a whole therefore avoids being biased towards a particular (incorrect) decision. Such approaches are relatively straightforward to translate to the machine learning domain. Numerous studies have demonstrated the improved accuracy of ensembles over single learner systems (Brown, 2010). Moreover, these same studies also show that the accuracy of the ensemble improves as the diversity of the ensemble improves (Kuncheva and Whitaker, 2003). However, as we observe later, formal mathematical guarantees for ensembles of learners, in particular ensembles of classifiers are scarce. Specifically, it is not well-understood when an ensemble will outperform the best single classifier.

### 1.1.2 Learning of High Dimension Datasets

Advances in computing technology enabled researchers to build datasets with the number of features (dimensionality) in the order of thousands. This increase in the dimensionality of the datasets has enabled researchers to explore interactions between features and hence, at least in principle, to build more accurate predictive models than previously was possible (Manyika et al., 2011).

Unfortunately, high dimensional datasets are especially challenging for machine learning (Durrant, 2014; Spruyt, 2014) due to the phenomenon dubbed as the "curse of dimensionality" (Bellman, 1970). Two extreme but typical cases are where firstly we have high-dimensional data with significantly more observations than the dimensionality of the data, in which case we run into time and space complexity issues. While on the other extreme, we have dimensionality of the data more than the number of observations, in which case we have problems making inferences and have bogus interactions between the features (Durrant, 2014).

One approach towards overcoming this challenge is by reducing the dimensionality of the dataset while retaining as much of the information from the dataset as possible. Dimensionality reduction can be achieved using deterministic dimensionality reduction methods such as Principal Components Analysis (PCA) or random dimensionality reduction approaches such as Random Projections (RP).

Using random dimensionality reduction approaches with ensemble learning approaches makes intuitive sense. Reducing the dimensionality helps to improve the computational efficiency and avoid the aforementioned mentioned time and space complexity issues at the cost of some accuracy loss. However, the accuracy loss can be recovered (and improved upon) through the ensemble approach on the many randomized low-dimensional projections of the data.

RP Ensembles has been shown to be successful examples of this. RP are computationally efficient, yet sufficiently accurate. Moreover, somewhat surprisingly it frequently works better than PCA despite the variability of single random projections (Bingham and Mannila, 2001). Durrant and Kabán (2014) showed that, for classification, combining several random projections could improve both classification performance and model stability, with theoretical guarantees on the ensemble classifier performance even when the number of training examples is far lower than the number of data dimensions.

An alternative random dimensionality reduction method is the Random Subspace method (RS) introduced by Ho (1998). RS is computationally more efficient than RP because RS merely involves selecting a subset of data feature indices randomly without replacement whereas RP requires a matrix-matrix multiplication. Additionally, RS projected datasets are more interpretable than RP projected datasets because RS retains a subset of the original features. However, despite the success of RS in many problem domains (Serpen and Pathical, 2009; Kuncheva et al., 2010; Ho, 1995; Breiman, 2001), there is very little theory to explain the effectiveness of RS.

## 1.2 Motivation and Research Questions

Ensemble classification with RS as a diversity generator scheme is an appealing research direction. RS has been shown to empirically perform very well and is used in many high-dimensional classification tasks (Kuncheva et al., 2010; Ho, 1998). Additionally, RS is easy to implement and significantly computationally cheaper than other dimensionality reduction techniques such as PCA and RP. However, despite the empirical results and extensive use of RS in ensemble learning, there are few theoretical explanations to said effectiveness, especially for classification tasks.

Theoretical insights into RS ensemble classifiers would also provide a more in-depth understanding of the theoretical foundations of other ensemble classifiers. While we have a sound theoretical basis for ensemble regression in terms of the bias-variance-covariance decomposition of their error (Ueda and Nakano, 1996), apart from specific cases such as Random Projection-Fishers Linear Discriminant (Durrant, 2013) and Negative Correlation Learning (Brown, 2010), we have little theory to explain the error decompositions in ensemble classifiers.

Additionally, insights grounded in the high-dimensional settings also help us understand the counter-intuitive nature of high-dimensional learning and possibly help in developing computationally efficient methods that would be helpful to learn tasks involving the increasingly high-dimensional data.

Ultimately, we hope that the insights gained would help us answer the key open question for ensemble classifiers, “When will an ensemble of weak learners outperform a single carefully-tuned learner?” (Durrant, 2013; Brown et al., 2005). With this motivation, some questions we are interested as follows. Note that when we talk of an ensemble, we refer to a classifier ensemble.

- What are the factors affecting the error in each ensemble method? i.e. How does the following affect the error and other performance metrics of the ensemble of classifiers in high-dimensional datasets (HDDS)?

  - Number of attributes in each classifier for Random Subspace method.
  - Selecting attributes for random subspace with non-uniform probability.
  - Number of members in an ensemble of classifiers.
  - Choice of the weight assignments for the combination schemes.
  - The use of different combiners methods, for instance, majority voting, weighted sum

- How does noise affect overall accuracy of the ensemble? What affects an ensemble's tolerance to noise (Schapire, 2013)?

- How does having negatively correlated errors affect the performance of the ensemble? What is the gain of having many weak or uncorrelated classifiers against having relatively fewer negatively correlated weak classifiers (Kuncheva et al., 2000; Brown et al., 2005)?

- Are there any "lucky" structures in the data — for example sparsity or regularity — that would help with classification? Is the performance of the classifier dependent on the representation of the data?

- What is a good measure of diversity in the ensemble and how does diversity affect the performance of the ensemble (Didaci et al., 2013; Kuncheva et al., 2000)?

- For a given problem, is there a "best" measure of diversity (Didaci et al., 2013)?

## 1.3 Organization of the Thesis

This thesis contains seven chapters including this chapter as well as supplementary materials containing supplementary proofs and figures not directly referenced in the thesis. The MATLAB and PYTHON source-codes used to produce the empirical results presented in this thesis are available at GitHub (`http://www.github.com/martianunlimited/phd-research/`), with a soft-copy of the source-codes attached to this thesis.

- **Chapter 1** introduces the problem and states some the research questions we would like to answer in this thesis.
- **Chapter 2** discusses the state of art and the gaps in our understanding.
- **Chapter 3** introduces the mathematical tools needed to derive the theorems and results presented in the later chapters.
- **Chapter 4** presents our investigation into the conditions for Johnson-Lindenstrauss Lemma-like norm-preservation-guarantees on random subspace projected data. Our main motivation behind this investigation is to provide the theoretical foundation for geometry-preservation for randomized dimensionality reduction (namely random subspace) that is independent of the concept class (Arriaga and Vempala, 1999). The implications of our investigation are far-reaching and goes beyond margin-based classifications, with implications also affecting non-machine learning applications such as sparse signal recovery using compressive sensing which would also be discussed in this chapter.
- **Chapter 5** presents our investigation on "flipping probability" defined as the probability that two vectors in $d$-dimensions with an angular separation of less than $\pi/2$ having an angular separation more than $\pi/2$ after projecting to a lower dimensional space (Durrant and Kabán, 2013). As noted by Durrant and Kabán (2013), results from this investigation

provides an upper bound on the generalization error of any linear classifier in a randomly projected space in the absence of a margin.

- **Chapter 6** investigates the performance of the ensemble when the classifiers are correlated. By establishing RS as a randomized dimensionality reduction, we can look at the RS process as an independent randomized diversity generation scheme for classifiers ensembles. Inspired by results from the social sciences and economics, we will investigate modelling the accuracy of a majority vote ensemble using a Polya-Eggenberger distribution and discuss the implications of the model. We will provide extensive empirical corroboration, and discuss other considerations affecting the accuracies of an ensemble classifier (e.g. feature/label noise, combination weights, training size).

- **Chapter 7**, we apply our findings to deep neural network image classification tasks. Taking inspiration from nature we propose "PseudoSaccade" and show how an ensemble of deep neural network classifiers with "PseudoSaccade" can give better image classification accuracy compared to a single view classification. Our approach is also highly robust to adversarial examples, unlike the original neural networks.

- **Chapter 8**, we summarise our findings and discuss future directions for this research.

# 2

# Background

**Summary** We begin our review by stating the classical intuitions behind ensemble learning and a review of the literature regarding ensembles of classifiers. We will then review high-dimensional learning and some dimensionality reduction techniques and show how dimensionality reduction strategies in high-dimensional learning can work well with ensemble learning. We will then introduce the work done for random projections ensemble classifiers in various literature (i.e. Durrant (2013); Durrant and Kabán (2013); Cannings and Samworth (2017); Arriaga and Vempala (1999)) to chart a framework for our analysis on random subspace ensembles. Finally, we will identify gaps in our current understanding and the challenges that make these gaps challenging to surmount.

## 2.1 State of the Art

### 2.1.1 Ensemble Classifiers

Ensemble classifiers are a "meta-learning" approach to machine learning, where rather than having a single classifier make the decision, the decision is made by training multiple "base" classifiers and combining their outputs such that the decision is the collective decision of the "committee". There are many different approaches for combining the ensemble member's outputs — we discuss several in section 2.1.2. Ensemble classifiers can be roughly categorised based on two features of the ensemble learner namely the method by which

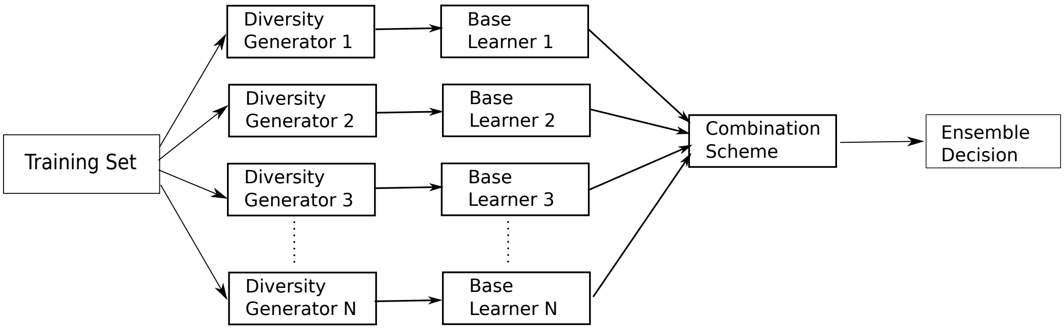

**Figure 2.1:** *Model of an Ensemble Learner*

diversity is induced between the base learners and the combination scheme for the base learners' outputs (Valentini and Masulli, 2002).

Figure 2.1 illustrates a general model of an ensemble classifier. In this figure, the components of ensemble classifier in this model are as follows:

- Training Set :- data set for training the base classifiers.
- Diversity Generator :- component generating the diversity in the base classifier outputs. Typically, this is achieved by manipulating the training data.
- Base Learner :- learners generated by one or more learning algorithms.
- Combination Scheme :- component responsible for combining the results from the ensemble members into a single decision rule.

In classification tasks, we can define $\mathcal{H}$ to be the hypothesis space representing the possible classifiers from a family of classifiers. That is $\mathcal{H}$ is a space in which points are functions, each of which is a possible outcome of the training data and learning algorithm. We learn $\hat{h} \in \mathcal{H}$ that minimizes the expected loss function $L(x_q)$, typically the misclassification rate (also known as the 0-1 loss) or the sum squared error between the output of the classifier and the actual output, given an observation $x_q \sim \mathcal{D}_{x|y}$, where $\mathcal{D}_{x|y}$ is the data generating distribution. Typically, $\mathcal{D}_{x|y}$ is unknown and has to be estimated from the 'training set' (a collection of examples) $\mathcal{T} \overset{i.i.d}{\sim} \mathcal{D}_{x|y}$ drawn identically and independently from the data generating distribution.

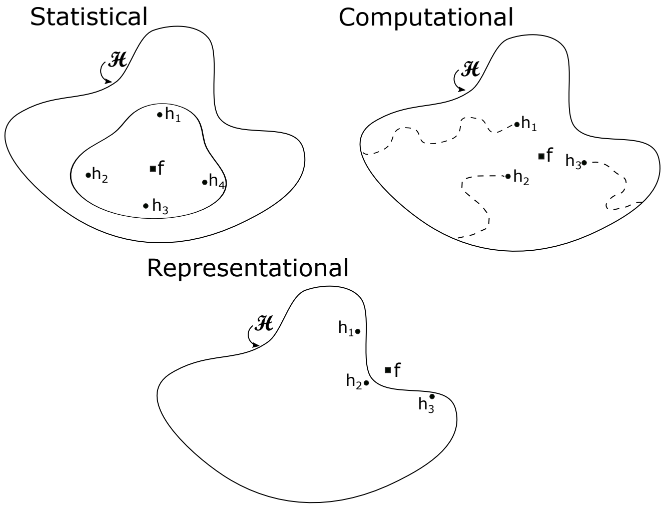

**Figure 2.2:** *Three fundamental reasons why an ensemble may work better than a single learner (adapted from (Dietterich, 2000a)). $\mathcal{H}$ represents the hypothesis space of all possible learners, $h1, h2, h3$ the individual base learners, and $f$ the decision rule output from the combination scheme.*

The ensemble learning approach to classification tasks comprises learning from the training data multiple $\hat{h}_i$, with $i \in \{1, \ldots, N\}$, and $N$ is the size of the ensemble. The ensemble learner then combines these individual classifiers using a combination scheme into a single decision rule that we hope will minimize the expected error for a given error function.

Empirical results have shown that ensemble classifiers are typically superior in terms of accuracy and robustness versus individual learners (Kuncheva, 2002). Figure 2.2 illustrates three intuitions as described by Dietterich (2000a) as to why an ensemble can be superior to an individual learner. The first comes from a statistical intuition whereby the average of the individual learners reduces the impact of learning a "bad" hypothesis that does not generalize well to data outside the training set (i.e. the "wisdom of crowds" discussed in the introduction (Chapter 1). Second is a computational intuition in that individual learners may converge to a local minimum of the loss, but an ensemble constructed from many starting points may provide a better approximation to the optimal learner with overall minimum loss. Last, is a representational intuition, where the hypothesis space (the space of all possible learners that the learning algorithm can generate) may not encompass the optimal learner, but the sum of the individual learners may expand the representable functions in the hypothesis space to give an aggregated learner that is closer to the optimal learner. For example, if the best decision boundary is quadratic, but $\mathcal{H}$ only contains linear classifiers, a piecewise linear classifier can better approximate the decision boundary than a single linear classifier. While these intuitions have not been established with formal theoretical foundations, these intuitions are widely accepted among researchers (Valentini and Masulli, 2002).

### 2.1.2 Combination Schemes in Ensemble Classifiers

As one might readily expect, the choice of the combination method in the combination scheme can significantly affect the overall accuracy of the ensemble learner (Leung and Parker, 2003). While in regression ensembles, we might typically combine the predictions using either a weighted average or median, there are many more methods in ensemble classifiers to combine the predictions of the outputs. Many of these methods are inspired by the electoral systems studied in political science and thus share the same shortcomings. Unsurprisingly, there is very little consensus as to which is the superior electoral system. Indeed, if there are more than two choices, it is impossible to satisfy a common set of reasonable conditions for any voting scheme simultaneously unless the decision is made by a dictator (Arrow, 1950).

An overview by Van Erp et al. (2002) categorised the combinations schemes into three categories depending on the output of the member classifiers. The first category of ensemble combination scheme is the vote-based schemes, where the base classifiers only provide a single class label as the decision from the classifier. The second category is described as rank-based schemes where the base classifiers provide a list of class labels in the order the classifier finds to be the most likely decision. Finally, the last category is score-based schemes where the classifier provides a list of class labels as well as a score representing the “confidence” the classifier has that the corresponding class label is correct or its estimate of the relative probabilities of the class labels.

To help elaborate on the different combination schemes, we will also borrow some terminology from the electoral systems. The “candidates” is the set of all possible class labels output from the classifiers and a “candidate” is a member element of the “candidates”. A vote is the “candidate” chosen as the top choice of the classifier, and the score is a numerical value representing the “confidence” of the classifier in that choice. A ranked list is an ordered list of candidates sorted according to the preference or the confidence that the “candidate” is the correct choice.

- Vote-based Schemes
  - Plurality: each classifier gives a vote for the class label, and the class label with the highest vote is the output of the ensemble. In political science, this is sometimes also known as “first past the post”. While this voting system makes intuitive sense, this system may result in a less preferred choice winning the vote due to something called the “spoiler effect” where the would-be winner shares the votes with a spoiler candidate, resulting in a less preferred candidate winning.
  - Majority: similar to plurality vote, except that if the top choice fails to obtain at least 50% of the votes, the ensemble does not produce an output. Note that most literature does not distinguish between plurality vote and majority vote and uses the definition given in plurality voting.
  - Amendment: Amendment voting compares two candidates and eliminates the candidate with the least vote. The winner is then pitted against the next available candidate and so on until the last remaining candidate is declared the winner. Note that amendment voting favours the candidate that is last to be added into the voting system, because if there are more than two classes, the preference may be non-transitive (e.g. like in a game of rock, papers, scissors).

  - Runoff: Runoff voting comprises of two rounds. The first round chooses two winners using the plurality vote rules from the choice of all the candidates. The second round chooses a final winner from the two winners of the first round.
  - Condorcet count: All candidates are compared in pairs with the winner in each pair awarded a point. The final winner is the candidate that was awarded the most points from each of the pairwise comparisons. This is sometimes known as round-robin in some literature (Fürnkranz, 2002).
- Rank-based Schemes
  - Borda count: This method was developed by Borda (1781) and uses the ranking from all voters and assigned a score based on the relative rank (typically $1/m$, where $m$ is the position of the candidate in the ranking list). We then compute the mean score of each candidate over all the voters. The classes are re-ranked by their mean score, and the top-ranked candidate is picked as the correct output. The Borda count can be seen as the analogue to Sum rule when the classifier confidence scores are unavailable.
  - Single transferable vote: This system is sometimes known in political science as "alternative voting". Under this system, the system attempts to find a winner through majority voting, if none of the candidates acquires the requisite 50% of the vote, the candidate with the lowest number of votes is eliminated and the candidate's vote given to the voter's next choice. The elimination of the candidate with the lowest number of votes is repeated until one of the candidates receives the requisite 50% of the votes. In machine language literature, this system is sometimes known as "Plurality with Elimination" (Leung and Parker, 2003; Leon et al., 2017) and a variant of this combination scheme exists known as "Anti-plurality" where rather than the classifiers voting for the most likely candidate,

    the classifiers votes against the candidate the classifier finds to be least likely to be correct.

- Score-based Schemes
    - Pandemonium: In pandemonium voting, the classifier provides a value representing the classifier's confidence in the prediction. The candidate class that receives the highest confidence among all the prediction is chosen as the output class (Selfridge, 1958). This combination rule is known as "max rule" in some literature (e.g. Kuncheva and Whitaker (2003)).
    - Sum rule: The ensemble sums the confidence of each of the candidates and chooses the candidate with the highest total. This method is functionally equivalent to the average vote rule used in some literature.
    - Product rule: This is similar to the sum rule, with the key difference where rather than summing the confidence score, the confidence scores are multiplied together. This combination rule severely penalises classes with a low confidence score. The product rule combination scheme is sometimes known as the geometric mean rule in some literature.

Ensemble classifiers commonly combine the decision using plurality vote (sometimes also known as hard-vote in software implementations) and sum rule (also known as soft-vote). While the other combination schemes (e.g. Borda count) are less commonly used, empirical results have shown that these combination schemes can give superior accuracy compared to majority-vote for specific applications (Riesen and Bunke, 2007; Ho et al., 1994; Domeniconi and Yan, 2004; Leon et al., 2017).

Figure 2.3 taken from Van Erp et al. (2002) shows that the accuracy of a bagged ensemble classifier varies depending on the ensemble combination schemes. The results in Van Erp et al. (2002) are consistent with the simulation by Kuncheva (2002). However, it is important to note that the authors noted that their results rely on several assumptions that may not be tenable in real-world situations (i.e. independence in classification error, identical accuracy).

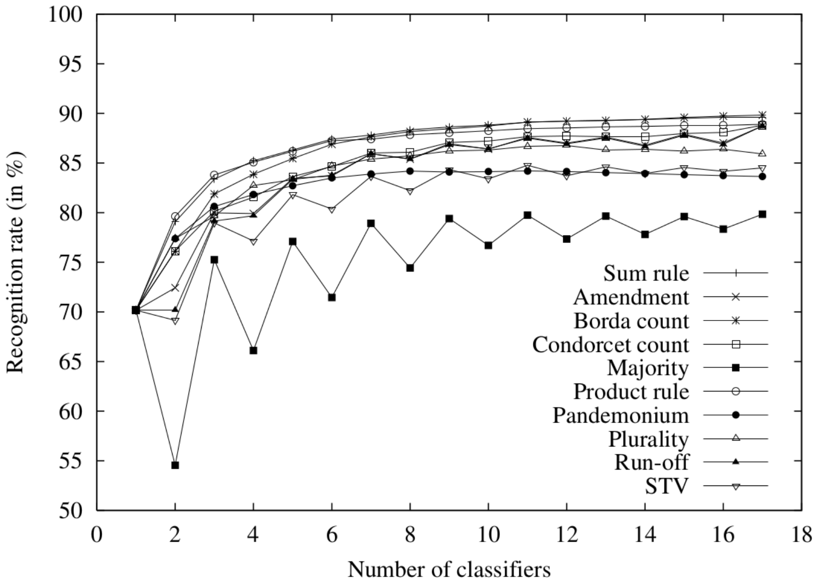

**Figure 2.3:** *Classification accuracy for handwriting of digits for different combination schemes taken from Van Erp et al. (2002)*

### 2.1.3 Measuring Diversity in Ensemble Classifiers

It is generally accepted ensemble learning takes inspiration from political science, where the ancient Greeks believe that the joint decision of the society is superior to that of an individual. This intuition was first explored in Condorcet's Jury Theorem (CJT) (Condorcet, 1785) which states that a group of voters which takes a majority vote between two alternatives of which exactly one is "correct", makes the correct decision with absolute certainty (with probability one) as the group size increases. CJT assumes that, the voters are sufficiently competent (correct at least more than half the time), and are independent, which however, have been shown to be simultaneously untenable (Dietrich, 2008). It is generally accepted however that, while we cannot guarantee that the decision would be "correct" with absolute certainty, the decision of the group would tend to be superior to that of a similar individual classifier. Many empirical results, including results from various authors (Kuncheva et al., 2000;

Tumer and Ghosh, 1996; Kuncheva and Whitaker, 2003) provided credence to this intuition, and it is generally accepted that for a given base classifier accuracy, the accuracy of the ensemble improves as the diversity of the base classifiers in it increases.

This emphasis on the diversity of the ensembles leads us to an open problem, how do we quantify diversity? To quote Zhou (2012),

> Though diversity is crucial, we still do not have a clear understanding of diversity; for example, currently, there is no well-accepted formal definition of diversity. There is no doubt that understanding diversity is the holy grail in the field of ensemble learning. (Zhou (2012) p. 100)

The comparison between algorithms and error analysis is made difficult because we do not have a common agreement on the various forms of diversity measures. At the moment, there is no unifying theory for all diverse ensembles or even all ensembles from a particular family. While there have been many empirical works on ensemble learning, it is difficult to definitively compare between the approaches as these results are generated with different datasets, different pre-processing, different classifiers, and different combination schemes. A theory is needed to understand the inner workings of what is going on to help researchers interpret and understand the results, to provide performance guarantees for different approaches, and (hopefully) to suggest new algorithms for particular problem settings.

Table 2.1 taken from Kuncheva and Whitaker (2003) summarises some of the different ways diversity is quantified. The definitions for some of the diversity measures are given in equations 2.1-2.5 with the definitions of $N_{00}, N_{01}, N_{10}$, and $N_{11}$ given in table 2.2

$$Q_{ij} = \frac{N_{11}N_{00} - N_{01}N_{10}}{N_{11}N_{00} + N_{01}N_{10}} \tag{2.1}$$

$$\rho_{ij} = \frac{N_{11}N_{00} - N_{01}N_{10}}{\sqrt{(N_{11}+N_{10})(N_{01}+N_{00})(N_{11}+N_{01})(N_{10}+N_{00})}} \tag{2.2}$$

| Name | Symbol | $\nearrow$ | Reference |
|---|---|---|---|
| Q-Statistics | $Q_{i,j}$ | - | Yule (1900) (Eqn: 2.1) |
| Correlation coefficient | $\rho_{i,j}$ | - | Sneath and Sokal (1963) (Eqn: 2.2) |
| Disagreement Measure | $D_{i,j}$ | + | Ho (1998) (Eqn: 2.3) |
| Double-fault measure | $DF_{i,j}$ | - | Giacinto and Roli (2001) (Eqn: 2.4) |
| Kohavi-Wolpert variance | $kw$ | + | Kohavi and Wolpert (1996) |
| Interrater agreement | $\kappa_{i,j}$ | - | Dietterich (2000b) (Eqn: 2.5) |
| Entropy measure | $Ent$ | + | Cunningham and Carney (2000) |
| Measure of difficulty | $\theta$ | - | Hansen and Salamon (1990) |
| Generalized diversity | $GD$ | + | Partridge and Krzanowski (1997) |
| Coincident failure diversity | $CFD$ | + | Partridge and Krzanowski (1997) |

**Table 2.1:** *Table of diversity measures taken from Kuncheva and Whitaker (2003). + indicates diversity increases with higher measure, and - indicates that the diversity increases with lower measure.*

$$D_{ij} = \frac{N_{01} + N_{10}}{N_{11} + N_{10} + N_{01} + N_{00}} \tag{2.3}$$

$$DF_{ij} = \frac{N_{00}}{N_{11} + N_{10} + N_{01} + N_{00}} \tag{2.4}$$

$$\kappa = \frac{2(N_{11}N_{00} - N_{01}N_{10})}{(N_{11} + N_{10})(N_{01} + N_{00}) + (N_{11} + N_{01})(N_{10} + N_{00})} \tag{2.5}$$

|  | $D_j$ Correct | $D_j$ Wrong |
|---|---|---|
| $D_i$ Correct | $N_{11}$ | $N_{10}$ |
| $D_i$ Wrong | $N_{01}$ | $N_{00}$ |

**Table 2.2:** *2×2 table of the relationship between the classifiers $D_i$ and $D_j$ , where $N_{xy}$ are counts of instances according to the relationship above.*

### 2.1.4 Results from social studies and economics

Despite the empirical evidence showing that diversity is important, many literatures on ensemble classifiers uses a binomial distribution which does not take in consideration the classifier diversity to model the accuracy of a majority vote ensemble classifier (e.g. Lam and Suen (1997); Whitaker and Kuncheva (2003); Kuncheva et al. (2003)).

Fortunately, because of the significant overlap between ensemble classifiers and voting theory and the enduring interest in the Social Sciences and Economics literature in CJT, we can exploit theoretical results from these areas. In the recent years, there has been significant number of results regarding CJT (e.g. Dietrich (2008); Paroush (1997); Dietrich and Spiekermann (2013); Karotkin and Paroush (2003)) and the implications on the competency of the collective decision of a group of decision maker when the assumptions of CJT are not satisfied. The results that are of most interest to us are by Ladha (1993), who showed that by using de Finetti's theorem, the assumption of independence for the voters could be relaxed to hold for weakly correlated voters. Moreover, Ladha (1995) later extended his result to show that, if we assume that each voter has identical competency, the number voters choosing the "correct" decision follows a Polya-Eggenberger Distribution (a generalized Beta-Binomial distribution that allows for negative valued shape parameters (Sen and Mishra, 1996)). This result was independently corroborated by Berg (1993) using a numerically equivalent variant of the distribution.

To the best of our knowledge, the results of these finding have not been explored before for machine learning applications. Although we were able to find a single paper implementing algorithm for an ensemble classifier based on the beta-binomial model by Ahn et al. (2007) and the application of that approach for a genomics problem by Ahn's student (Fazzari, 2007), the theoretical implications of the results from the social sciences remains largely unexplored for machine learning.

### 2.1.5 Error Decomposition of Ensemble Classifiers

While we have a good theoretical foundation for the error decomposition in ensemble regression in the bias-variance-covariance error decomposition, the theoretical foundations founded in ensemble classifiers cannot be transferred directly to ensemble classifiers. To see why this is so, we consider the error decomposition of an ensemble regression with a squared error loss. We let $f_i$ be the output of the base learner $\hat{h}_i$ and $f_{ens}$ be the output of the ensemble. We also let $N$ be the size of the ensemble, and $y$ be the true label. The mean squared error of the uniform weighted ensemble regression can then be written as,

$$\begin{aligned} \mathrm{E}\left[(f_{ens}-y)^2\right] = &\left(\frac{1}{N}\sum_i^N \mathrm{E}\left[f_i - y\right]\right)^2 + \left(\frac{1}{N^2}\sum_i^N \mathrm{E}[(f_i-y)^2]\right) \\ &+ (1-\frac{1}{N})\sum_i^N\left(\frac{1}{N(N-1)}\sum_{j\neq i}\mathrm{E}[(f_i-\mathrm{E}[f_i])(f_j-\mathrm{E}[f_j])]\right) \end{aligned}$$

The first term is said to be the average bias of the member regression, the second term the average variance of the ensemble learners and the third term the average covariance of the ensemble member. The error is therefore minimized when the bias and variance terms are minimal and the covariance terms maximally negatively correlated. However, in classification problems, both $f_i$ and $y$ are non-ordinal values, and therefore the concept of variance and covariance is difficult to define. Moreover, the loss functions (e.g. zero-one loss, ReLU, logistics) used in classification algorithms usually cannot be decomposed into functions involving the bias and variance. To quote Brown et al. (2005),

> The harder question can therefore be phrased as, “How can we quantify diversity when our predictors output non-ordinal values and are combined by a majority vote?” Taking all these into account, there is simply no clear analogue of the bias-variance-covariance decomposition when we have a zero-one loss function. We instead have a number of highly restricted theoretical results, each with their own assumptions that are probably too strong to hold in practice. (Brown et al. (2005) p. 7)

Krogh and Vedelsby (1995) provided a proposed framework for error decomposition of ensemble classifier using what was described as ambiguity decomposition. In ambiguity decomposition, the error is decomposed into the (weighted) average error and the deviation of the individual classifier to the ensemble output (ambiguity). The equation below shows the squared error at a single data point.

$$\mathrm{E}\left[(f_{ens} - y)^2\right] = \left(\frac{1}{N}\sum_{i}^{N}(f_i - y)^2\right) + \underbrace{\frac{1}{N}\sum_{i}^{N}(f_i - f_{ens})^2}_{\text{Ambiguity Term}}$$

Brown and Wyatt (2003) showed the relationship between ambiguity decomposition and the bias-variance decomposition. Moreover, Brown (2004) also demonstrated how this framework could explicitly be used to control the accuracy-diversity trade-off in negative correlation learning. However, it was noted by Zhou (2012) that the variance term exists in both the error term and the ambiguity term, indicating that it is difficult to maximise the ambiguity of the classifiers without also affecting the bias term.

Another possible framework for the error decomposition is by Brown and Kuncheva (2010), in which the authors further decomposed the "ambiguity" term into what is called "good" diversity and "bad" diversity, defined as the disagreement between the ensemble decision and the individual classifier decision. The equation below shows the "good" and "bad" diversity decomposition, with $\boldsymbol{x}^+$ the data points where the ensemble classified correctly, and $\boldsymbol{x}^-$ the data point where the ensemble classified incorrectly.

$$\mathrm{E}\left[L(f_{ens} - y)\right] = \int_{\boldsymbol{x}} L(f_i - y) + \underbrace{\int_{\boldsymbol{x}^-}\frac{1}{N}\sum_{i}^{N} L(f_i - f_{ens})}_{\text{“Bad” Diversity}} - \underbrace{\int_{\boldsymbol{x}^+}\frac{1}{N}\sum_{i}^{N} L(f_i - f_{ens})}_{\text{“Good” Diversity}}$$

### 2.1.6 Diversity Generation in Ensembles

Despite the lack of agreement in the definition of diversity, researchers are not discouraged from developing algorithms to increase the diversity in the ensemble. Brown (2010) categorised diversity generation into two categories, "implicit" diversity and "explicit" diversity. "Explicit" diversity generation are said to be methods where the diversity is measured and actively "encouraged", whereas "implicit" diversity generation "assumes" that the random process would create diversity in the ensemble.

In the survey on ensemble learning by Sewell (2011), one of the standard approaches for ensemble learning is "Bagging" (bootstrap aggregation learning).

Bagging as introduced by Breiman (1996) generates multiple predictors that exploit the diversity generated by taking multiple bootstrap replicates of the training data, where the bootstrap replicate is a random sample with replacement of the original dataset. These predictors are then combined using some aggregation method (e.g. plurality voting). Observe that Bagging is an example of a method that exploits "implicit" diversity generation according to the definition given above.

```
$n \leftarrow$ size of the training set,
$N \leftarrow$ size of the ensemble
$\mathcal{T} := \{(\boldsymbol{x}_1|y_1), \ldots, (\boldsymbol{x}_n|y_n)\}$ be the representing the training set with observations $\boldsymbol{x}_i$ and label $y_i$.
for $i \leftarrow 1$ to $N$ do
    - Create training set $\mathcal{T}_i$ by sampling from $\mathcal{T}$, $m \leq n$ items uniformly at random with replacement.
    - Learn $h_i$ using this training set $\mathcal{T}_i$, and add it into the ensemble.
end for
Combine the output of $h_i(\boldsymbol{x})$ using some combination scheme, (e.g. sum rule)
    $h_{ens}(\boldsymbol{x}) = \frac{1}{N} \sum h_i(\boldsymbol{x})$
```

**Algorithm 2.1:** *Algorithm for Bagging taken from Brown (2010)*

An alternative to Bagging is what is called "Boosting" introduced by Schapire (1990). In Boosting, the training set is resampled non-uniformly rather than uniformly. While there is a large family of Boosting algorithms, one of the more investigated and successful variant is AdaBoost (Adaptive Boosting). In AdaBoost, the training set is weighted such that the examples that are misclassified by previous ensemble members are sampled with higher probability as the training procedure advances. AdaBoost can be considered as an example of an ensemble method that exploits "explicit" diversity generation in that the approach of AdaBoost (also known as residue importance sampling) adaptively reduces the correlation between the errors of subsequent classifiers and earlier ones, thereby improve the accuracy of the ensemble.

```
$n \leftarrow$ size of the training set,
$N \leftarrow$ size of the ensemble
$\mathcal{T} := \{(\boldsymbol{x}_1|y_1), \ldots, (\boldsymbol{x}_n|y_n)\}$ be the representing the training set with observations $\boldsymbol{x}_i$ and label $y_i$.
Define an initial probability distribution $\mathcal{D}_1(m)$ representing the sampling probability of training example $m$ from $\mathcal{T}$. e.g. $\mathcal{D}_1(n) = \frac{1}{n}, \forall m \in [1, n]$
for $i \leftarrow 1, N$ do
    - Create new training set $\mathcal{T}_i$ by sampling with replacement from $\mathcal{T}$, $m \leq n$ items according to probability distribution $\mathcal{D}_i$.
    - Learn $h_i$ using this training set $\mathcal{T}_i$, and add it into the ensemble.
    - Calculate $w_i$ according to accuracy $acc_i$ of $h_i$, e.g. $(w_i = logit(acc_i))$
    - Update $\mathcal{D}_{i+1}$ such that $\mathcal{D}_{i+1}(j)$ is increased if instance $j$ is misclassified and decreased otherwise.
    - Normalize $\mathcal{D}_{i+1}$ so that $\mathcal{D}_{i+1}$ is a distribution
end for
Combine the output of $h_i(\boldsymbol{x})$ using some combination scheme, (e.g. weighted majority vote)
    $h_{ens}(\boldsymbol{x}) = \frac{1}{\sum w_i} \sum w_i h_i(\boldsymbol{x})$
```

**Algorithm 2.2:** *Algorithm for "AdaBoost" taken from Brown (2010)*

While Bagging and Boosting are popular diversity generation methods for ensembles, empirical results have shown that these ensemble methods may not be suitable for high-dimensional learning (Piao et al., 2015). These findings are not surprising considering that subsampling the training set usually exacerbates the "curse of dimensionality" especially in data with small number of samples. Additionally, Boosting can be very sensitive to mislabelled examples (Zhou, 2012). However, there are Boosting algorithms specifically designed to be robust against label noise such as rBoost (Bootkrajang and Kabán, 2013).

### 2.1.7 High-Dimensional Learning

While high-dimensional datasets introduce a significant challenge to learning algorithms, namely the “Curse of Dimensionality” referenced in section 1.1.2, high-dimensional settings also come with some useful result that can be leveraged to help learning tasks. Donoho et al. (2000) coined the term “Blessings of Dimensionality” in his talk referring to results in concentration measures that provide high probability guarantees in the high-dimensional settings. Since 2014, the term “Blessing of Dimensionality” has increasingly appeared in literature, sometimes referring to concentration of measures (Gorban et al., 2016; Kucheryavskiy, 2018; Anderson et al., 2014), but also referring to the improved discriminative ability in high-dimensional representation (Liu et al., 2017; Lin et al., 2018; Pereda et al., 2018). While the results from the concentration of measures have been extensively applied for Random Projections (Durrant and Kabán, 2013; Matoušek, 2008), to the best of our knowledge there is no literature using results from the concentration of measure to provide probabilistic guarantees for Random Subspace (RS) projections.

One approach towards overcoming the curse of dimensionality is dimensionality reduction, which is to find a projection that projects a data point $x \in \mathbb{R}^d$ onto a $k$-dimensional subspace while retaining as much information from the data as possible. Borrowing from the taxonomy introduced by Brown (2010) referenced in section 2.1.6, we can categorize dimensionality reduction methods into “explicit” and “implicit” methods. Here we define explicit dimensionality reduction as methods that actively measure the distortion resulting from the projection and choose the projection that minimizes said distortion, while implicit dimensionality reduction is probabilistic methods that project the data down into the lower dimensional subspace without actively measuring the distortion caused by the projections. Examples of explicit dimensionality reduction methods include Principal Component Analysis, Independent Component Analysis, and Isomap (Sorzano et al., 2014). Another interesting example of explicit dimensionality reduction is feature selection, where the algorithm chooses $k$ features that are "most informative" in the data without transforming the features. This approach can be thought of as the "explicit" analogue to the RS method. Feature selection has been shown to be an effective approach for dimension reduction in some high-dimensional data (Nogueira and Brown, 2015).

### 2.1.8 Random Projection (RP)

Random projection (RP) is a randomised dimensionality reduction method that projects a data point $x \in \mathbb{R}^d$ onto a $k$-dimensional subspace with the subspace typically either chosen uniformly at random from all possible such subspaces of dimension $k$ in $\mathbb{R}^d$ or is the span of $k$ vertices of a centred hypercube chosen uniformly at random with replacement from all $2^d$ such vertices. In the implementation for a single RP, we generate a $k \times d$ matrix of values sampled from such a zero-mean symmetric sub-Gaussian distribution, and then left multiplies the data point with this RP matrix, with the same RP matrix being used for each data point in a training set of observations.

The RP method has its roots in geometric functional analysis and entered the Machine Learning and KDD communities via Theoretical Computer Science, in particular, seminal papers by Indyk and Motwani (1998) and Arriaga and Vempala (1999). RP has found many successful applications (Bingham and Mannila, 2001; Venkatasubramanian and Wang, 2011) and the theoretical foundations of RP are by now quite well understood (Dasgupta and Gupta, 2003; Matoušek, 2008; Indyk, 2001).

A key theoretical result regarding RP, widely used in theoretical analyses and as heuristic justification for the application of RP, is the following Johnson-Lindenstrauss Lemma (JLL):

**Proposition 2.1** (Johnson and Lindenstrauss, 1984)**.** *Let $\epsilon \in (0,1)$. Let $N, k \in \mathbb{N}$ such that $k \geq C\epsilon^{-2} \log N$, for a large enough absolute constant $C$. Let $V \subseteq \mathbb{R}^d$ be a set of $N$ points. Then there exists a linear mapping $R : \mathbb{R}^d \to \mathbb{R}^k$, such that for all $u, v \in V$:*

$$(1-\epsilon)\|u-v\|_2^2 \leq \|Ru - Rv\|_2^2 \leq (1+\epsilon)\|u-v\|_2^2$$

JLL has been extensively studied and surveyed (Matoušek, 2008). Unfortunately, RP is in general computationally much more expensive than RS, and of course, it does not preserve the original features. The time complexity to generate the projection matrix is $O(kd)$, and to extract the projected data from the full data requires a matrix-matrix multiplication, which is $O(kdN)$ in general. Although there are several approaches that consider increasing the sparsity of the projection (Achlioptas, 2001; Ailon and Chazelle, 2009; Kane and Nelson, 2014) to improve the hidden constants in the matrix multiplication, in practice, this is still costly for large or very high-dimensional datasets. For RP matrices with ±1 entries, Ailon and Liberty (2009) give an $O(Nd \log k)$ algorithm provided $k < \sqrt{d}$. For these and similar matrices such as those in (Achlioptas, 2001), one can also use Mailman's Algorithm (Liberty and Zucker, 2009) which, for a one-off pre-processing cost of $O(kd)$, speeds up the matrix-matrix multiplication by a factor of $O(\log d)$. However, our experience is that this approach is not as fast in practice as RS and, in particular, it is very memory hungry, and the data projection is slower in practice in typical use-cases. Finally, Ailon and Chazelle (2009) gave an $O(d \log d + N(d \log k + k^2))$ algorithm using a randomized Hadamard transformation to precondition the data so that, with high probability, it is regular. One of the key results for JLL is the result by Arriaga and Vempala (1999) which used the results JLL guarantees to upper-bound the generalisation error for margin-based classifiers.

Durrant and Kabán (2010) introduced the "flipping probability" of a pair of randomly projected vectors as the probability that two vectors in $d-$dimensional Euclidean space $\boldsymbol{m}, \boldsymbol{n} \in \mathbb{R}^d$ which are separated in $\mathbb{R}^d$ by an angle $\theta \in [0, \pi/2]$

have angular separation $\theta_R > \pi/2$ following a random projection. Later work by Kabán and Durrant (2017) shows that the “flipping probability” is a useful tool to capture the geometric structure that makes a classification problem “easy” in that it requires a relatively low amount of sample size to guarantee good generalisation and does not require a margin. However, Durrant (2013) noted that methods used to derive the “flipping probability” rely on the rotational invariance and therefore cannot be used on projections that are not rotationally invariant (e.g. RS).

### 2.1.9 Random Subspace Projection (RS)

Random Subspace method (RS) was first introduced by Ho (1998), where an ensemble of decision trees employing several sets of RS projected data was used for a classification problem. RS as an ensemble method has shown good results with many learning algorithms such as support vector machines, Tao et al. (2006), linear classifiers Skurichina and Duin (2002), $k$-nearest neighbour Ho (1995) and also on a variety of data sets from different problem domains (e.g. Kuncheva et al. (2010); Li and Zhao (2009); Lai et al. (2006)). Additionally, RS has several practical advantages over RP. In particular, unlike RP, it retains the original data features. Also, unlike RP, it can be used even if the data dimension $d$ is not fixed or is not known *a priori*, e.g. by using reservoir sampling (Vitter, 1985) on the feature indices. It has very low time complexity compared to RP, namely $O(d)$ (or $O(d \log d)$ using reservoir sampling) typically to generate a subset of indices to be sampled, and $O(N)$ to construct the projected dataset. Finally, we note also that scalable parallel approaches for sampling from very large and streaming datasets have recently been devised (Meng, 2013).

Formally, RS is a randomised dimensionality reduction method that projects a data point $x \in \mathbb{R}^d$ onto the subspace spanned by $k$ canonical basis vectors $e_j = (e^{(j1)}, e^{(j2)}, \dots, e^{(jd)})^T$, $j = \{1, 2, \dots, k\}$, where $e^{(ji)} = 1$ if $i = j$ and zero otherwise. The RS basis is chosen uniformly at random from all $\binom{d}{k}$ possible such subspaces of dimension $k$. In the implementation for a single RS one simply selects a subset of $k$ feature indices without replacement, uniformly at random from all such subsets of size $k$, and then discards the values of the remaining $d - k$ features with the same $k$ feature indices being used for each data point in a set of observations.

Loupes (2014) provided theory for a variant of RS known as Random Forest (Breiman, 2001), in which the author stated that building an ensemble reduces the variance of the class probability estimate. However, it may be important to note that the author's results assume squared error loss and an error estimate modelled using standard normal distribution.

## 2.2 Gaps in Current Understanding and Our Contributions

One of the key shortcomings we identified and liked to address in our thesis is the lack of theory for RS projections for ensemble learning. While there are numerous empirical results demonstrating the effectiveness of RS and RS-like ensemble learning, there is very little theory to explain the effectiveness of RS projections for learning. Many of the results in literature usually show improvements empirically for a specific problem domain with little evidence that the approach can be used in other problem domains. In chapter 4, we investigate the conditions for "norm-preservations" in RS projections. As far as we know, there are no known non-trivial guarantees for norm preservations in RS projections. Insights into this provides us with the theoretical basis for RS in algorithms that are based on geometry of the data (e.g. clustering, nearest-neighbour) and margin-based classifications.

Work by Kabán and Durrant (2017) shows that the "flipping probability" can be used to give an upper-bound to the generalization error in the absence of a margin. However, as noted by Durrant (2013), the proofs involving for the "flipping probability" on RP takes advantage of the rotational invariance nature of the RP projection and therefore cannot be transferred directly to projections that are not rotational invariant (e.g. RS). In chapter 5, we tackle this problem and provide an upper-bound to the flipping probability for RS projections which does not depend on rotational invariance. We then discuss the implications of the "flipping probability" in a RS ensemble taking into consideration that RS is an independent randomized diversity generation scheme.

In spite of the empirical evidence showing that diversity is important to the ensemble accuracy, accuracy models for majority vote are typically based on the binomial model which assume independence of votes and does not take in account diversity of the classifiers (Lam and Suen, 1997; Whitaker and Kuncheva, 2003; Kuncheva et al., 2003). Leveraging on the work from the social sciences, we investigate the majority vote accuracy of an ensemble of classifiers of correlated classifiers. We evaluate the accuracy-diversity trade-offs using a Polya-Eggenberger model (Ladha, 1995; Berg, 1993) and compare the model to extensive empirical results. We then discuss the implications of the model and provide extensive empirical corroboration for the model.

We provide empirical corroboration for our main results with synthetic and real-world high-dimensional data. Based on our results and our theory, we propose a method of building ensembles for Deep Neural Network (DNN) image classifications tasks using multiple RS projections and a single DNN to improve on the classification accuracy — without needing to retrain the neural network. Our approach shows improved accuracy versus existing non-ensemble approaches, and is highly robust to adversarial examples, unlike the original neural networks.

# 3

# Mathematical Tools

## 3.1 Linear Algebra

In this section, we will introduce some key results related to real-valued vector spaces and matrices. ‘

**Definition 3.1** (Matrices)**.** *Let $\mathbb{F}$ be a field and $m, n \in \mathbb{N}$. An $m \times n$ matrix $\boldsymbol{A}$ over $\mathbb{F}$ is defined as an array with $m$ rows and $n$ columns of numbers in $\mathbb{F}$. If $m = n$ (that is to say there the number of rows and columns are the same) the matrix is said to be a square matrix. We denote the entry in row $i$ and column $j$ as $A_{i,j}$. We use $A_{i,:}$ to denote every entry in the $i$-th row, and similarly, we use $A_{:,j}$ to denote every entry in the $j$-th column.*

These results hold for all fields, however in the following chapters, we will consider primarily on the real value fields, $\mathbb{F} = \mathbb{R}$, real-valued vectors $\mathbb{V} = \mathbb{R}^d$ and real-valued matrices $\mathbb{M} = \mathbb{R}^{m \times n}$

**Definition 3.2** (Vectors)**.** *Let $\mathbb{F}$ be a field and $d \in \mathbb{N}$. A vector is an ordered array over a field $\mathbb{F}^d$. Vectors are typically denoted with boldface lower-case letters, such as $\boldsymbol{x}$. A vector can also be thought of as a matrix with one column. The elements of the vector are identified using by the name of the vector followed by a subscript. For example, $x_1$ is the first element of the vector $\boldsymbol{x}$, and $y_3$ is the third element of the vector $\boldsymbol{y}$.*

To access multiple elements in the vector, we can define a set and subscript the vector with the set. For example, to access elements $\{1, 4, 7\}$, we define $s = \{1, 4, 7\}$ and denote $\boldsymbol{x}_s$. We use the $-$ symbol to complement the index, for example $\boldsymbol{x}_{-s}$ accesses all elements of $\boldsymbol{x}$ except elements $\{1, 4, 7\}$. We can compactly define a vector by defining the either explicitly defining the elements in the vectors, or to implicitly defining the vector. For example, the following are two ways of defining the same vector.

- $x_0 = 1, x_i = 0, \forall i \in [2, d]$
- $\boldsymbol{x} = [1, 0, \ldots, 0]$

**Definition 3.3** (Matrix Transpose)**.** *Let $\boldsymbol{A}$ be an $m \times n$ matrix. The transpose of matrix $\boldsymbol{A}^T$ is an $n \times m$ matrix with the row and column entries swapped, that is to say $(A^T)_{i,j} = A_{j,i}$.*

$$\boldsymbol{A} = \begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,1} & A_{2,2} & A_{2,3} \end{bmatrix} \qquad \boldsymbol{A}^T = \begin{bmatrix} A_{1,1} & A_{2,1} \\ A_{1,2} & A_{2,2} \\ A_{1,3} & A_{2,3} \end{bmatrix}$$

**Definition 3.4** (Matrix Addition and Multiplications)**.** *The matrix addition $\boldsymbol{A} + \boldsymbol{B}$ an element-wise operator on the matrix and exist if and only if $\boldsymbol{A}$ and $\boldsymbol{B}$ has the same number of rows and columns. Formally, let $\boldsymbol{A}$ and $\boldsymbol{B}$ be a $m \times n$ matrix, the sum $\boldsymbol{C} = \boldsymbol{A} + \boldsymbol{B}$ will be an $m \times n$ matrix with entries of $C_{i,j} = A_{i,j} + B_{i,j}$.*

The matrix product $\boldsymbol{AB}$ exist only if the number of columns in matrix $\boldsymbol{A}$ equals the number of rows in matrix $\boldsymbol{B}$. Formally let $\boldsymbol{A}$ be an $m \times n$ matrix, and $\boldsymbol{B}$ be an $n \times o$ matrix, the product $\boldsymbol{C} = \boldsymbol{AB}$ will be an $m \times o$ matrix with entries of $C_{i,j} = \sum_{k=1}^{n} A_{i,k} B_{k,j}$. Observe that assuming that the matrix operation is valid, the matrix operations satisfy the following axioms:

- $\boldsymbol{A} + \boldsymbol{B} = \boldsymbol{B} + \boldsymbol{A}$ (Commutative Addition)
- $(\boldsymbol{A} + \boldsymbol{B}) + \boldsymbol{C} = \boldsymbol{A} + (\boldsymbol{B} + \boldsymbol{C})$ (Associative)
- $(\boldsymbol{AB})\boldsymbol{C} = \boldsymbol{A}(\boldsymbol{BC})$ (Associative)
- $\boldsymbol{A}(\boldsymbol{B} + \boldsymbol{C}) = \boldsymbol{AB} + \boldsymbol{AC}$ (Distributive)
- $(\boldsymbol{A} + \boldsymbol{B})\boldsymbol{C} = \boldsymbol{AC} + \boldsymbol{BC}$ (Distributive)

In general, matrix products are not commutative (i.e. $\boldsymbol{AB} \neq \boldsymbol{BA}$), and the matrix product $\boldsymbol{AB}$ commutes if and only if $\boldsymbol{A}$ and $\boldsymbol{B}$ are simultaneously diagonalizable.

**Definition 3.5** (Vector Addition)**.** *The vector addition between vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d$ is the element-wise sum of the elements in $\boldsymbol{u}$ and $\boldsymbol{v}$. Formally, if $\boldsymbol{w} = \boldsymbol{u} + \boldsymbol{v}$, then $w_i = u_i + v_i, \forall i \in [1, d]$. Geometrically, we can interpret $\boldsymbol{u}$ as a directed vector in a d-dimensional space, and the vector addition $\boldsymbol{u} + \boldsymbol{v}$ can be interpreted as placing the tail of vector $\boldsymbol{v}$ to the head of vector $\boldsymbol{u}$ as denoted in the illustration in Figure 3.1.*

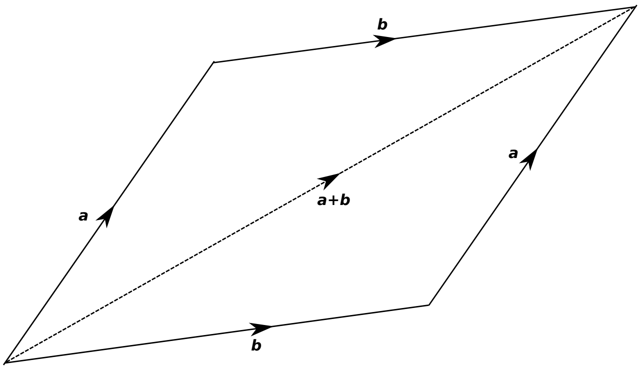

**Figure 3.1:** *A visual representation of a vector addition*

Additionally, for all vectors $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in \mathbb{V}$ and scalars $a, b$ in $\mathbb{F}$, the operation satisfies the following axioms:

- Commutativity: $\boldsymbol{u} + \boldsymbol{v} = \boldsymbol{v} + \boldsymbol{u}$
- Associativity: $(\boldsymbol{u} + \boldsymbol{v}) + \boldsymbol{w} = \boldsymbol{u} + (\boldsymbol{v} + \boldsymbol{w})$
- Additive Identity: $\boldsymbol{u} + \boldsymbol{0} = \boldsymbol{u}$ and $\boldsymbol{u} + (-\boldsymbol{u}) = \boldsymbol{0}$
- Distributivity: $a(\boldsymbol{u} + \boldsymbol{v}) = a\boldsymbol{u} + a\boldsymbol{v}$

**Definition 3.6** (Inner products)**.** *The dot product, also known as the inner product, is a product between two vectors and results in a scalar quantity. Formally, the scalar product $c = \boldsymbol{u} \cdot \boldsymbol{v}$ where $\boldsymbol{u} \in \mathbb{R}^d$ and $\boldsymbol{v} \in \mathbb{R}^d$ is given by $c = \sum_{i=1}^{d} u_i v_i$.*

Note that $\boldsymbol{a} \cdot \boldsymbol{b}$ is sometimes written using the matrix notation $\boldsymbol{a}^T \boldsymbol{b}$ or the inner product notation $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$. To keep the notation compact, we use the matrix notation in our proofs; however, we would occasionally use the inner product notation in our text to improve the readability of the statements.

Observe that a dot product has the following properties, namely that for all $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^d$:

- Positivity: $\langle \boldsymbol{u}, \boldsymbol{u} \rangle \geq 0$
- Definiteness: $\langle \boldsymbol{u}, \boldsymbol{u} \rangle = 0 \iff \boldsymbol{u} = \boldsymbol{0}$
- Additivity: $\langle \boldsymbol{u} + \boldsymbol{v}, \boldsymbol{w} \rangle = \langle \boldsymbol{u}, \boldsymbol{w} \rangle + \langle \boldsymbol{v}, \boldsymbol{w} \rangle$
- Homogeneity: $\langle a\boldsymbol{u}, \boldsymbol{v} \rangle = a \langle \boldsymbol{u}, \boldsymbol{v} \rangle$
- Conjugate Symmetry: $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \overline{\langle \boldsymbol{v}, \boldsymbol{u} \rangle}$

Note: For real value fields, conjugate symmetry implies commutativity. Geometrically, the dot product between $\boldsymbol{u} \cdot \boldsymbol{v}$ can be interpreted as the product of the projection of vector $\boldsymbol{u}$ on vector $\boldsymbol{v}$ with the vector $\boldsymbol{v}$. Figure 3.2 illustrate this geometrical interpretation.

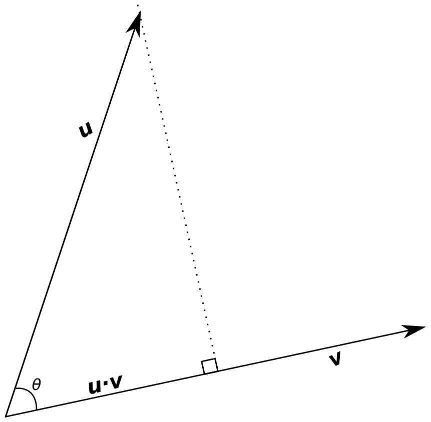

**Figure 3.2:** *Geometric representation of vector dot products*

**Definition 3.7** (Orthogonal vector)**.** *Vectors $\boldsymbol{u}, \boldsymbol{v}$ is said to be orthogonal to each other if $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = 0$. Geometrically, if vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ are both not $\boldsymbol{0}$, we can interpret the two vectors as perpendicular to each other.*

**Definition 3.8** (Hadamard Product)**.** *The Hadamard dot product($\odot$), is an element-wise product of the vectors. Formally, the Hadamard dot product*

$$\boldsymbol{w} = \boldsymbol{u} \odot \boldsymbol{v}, \boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d$$

*where* $w_i = u_i v_i$*.*

Observe that the Hadamard product has the following properties namely that for all $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^d$:

- Commutative: $\boldsymbol{u} \odot \boldsymbol{v} = \boldsymbol{v} \odot \boldsymbol{u}$
- Associative: $(\boldsymbol{u} \odot \boldsymbol{v}) \odot \boldsymbol{w} = \boldsymbol{u} \odot (\boldsymbol{v} \odot \boldsymbol{w})$
- Distributive: $\boldsymbol{u} \odot (\boldsymbol{v} + \boldsymbol{w}) = \boldsymbol{u} \odot \boldsymbol{v} + \boldsymbol{u} \odot \boldsymbol{w}$

We would define $\boldsymbol{u}^2$ as $\boldsymbol{u} \odot \boldsymbol{u}$ and more generally $\boldsymbol{u}^n$ with $n \in \mathbb{N}$ as $\boldsymbol{u} \odot \boldsymbol{u} \cdots \odot \boldsymbol{u}$ $n$-times. The Hadamard dot product is related to the dot product by $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \sum_{i=1}^{d} (\boldsymbol{u} \odot \boldsymbol{v})_i$. However, to the best of our knowledge, there is no intuitive geometric interpretation for the Hadamard dot product.

### 3.1.1 Matrix Operations

**Definition 3.9** (Matrix Trace)**.** *The trace of a square matrix* $\boldsymbol{A}$ *is the sum of the diagonal elements. Formally, the trace of an* $n \times n$ *matrix* $\boldsymbol{A}$

$$Tr(\boldsymbol{A}) = \sum_{i=1}^{n} A_{i,i}$$

.

**Definition 3.10** (Matrix singular value decomposition)**.** *A* $m \times n$ *real-valued matrix* $\boldsymbol{A}$ *can be written as the product of matrices* $\boldsymbol{U}$*,* $\boldsymbol{D}$*,* $\boldsymbol{V}^T$*, where* $\boldsymbol{U}$ *is a* $m \times m$ *orthogonal matrix,* $\boldsymbol{V}$ *is a* $n \times n$ *orthogonal matrix and* $\boldsymbol{D}$ *is a* $m \times n$ *matrix with values only along the main diagonal (i.e.* $D_{i,j} = 0$ *where* $i \neq j$*). Formally,*

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$$

*such that* $\boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{I}_m$ *and* $\boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{I}_n$*. The values along the diagonal of* $\boldsymbol{D}$ *are the singular values of matrix* $\boldsymbol{A}$*.*

*The rank of the matrix equals the number of non-zero diagonal elements of* $\boldsymbol{D}$*. A* $m \times n$ *matrix* $\boldsymbol{A}$ *is said to be full rank if Rank(*$\boldsymbol{A}$*)=*$\min(m, n)$*.*

**Definition 3.11** (Matrix Inverse)**.** *The inverse of a* $m \times m$ *square matrix* $\boldsymbol{A}$ *is the matrix* $\boldsymbol{A}^{-1}$ *such that*

$$\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}$$

*where $\boldsymbol{I}$ is the identity matrix. Note that only full-rank matrices (i.e. Rank($\boldsymbol{A}$) = m) has an inverse.*

**Definition 3.12** (Moore-Penrose Inverse)**.** *The Moore-Penrose inverse $\boldsymbol{A}^+$ (sometimes known as the pseudo-inverse) of a matrix $\boldsymbol{A}$ is the generalization of the inverse matrix $\boldsymbol{A}^{-1}$. For an $m \times n$ matrix $\boldsymbol{A}$, if $m \leq n$ and rank($\boldsymbol{A}$) $= m$, the Moore-Penrose inverse can be defined as*

$$\boldsymbol{A}^+ = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T$$

*and if $n \leq m$ and rank($\boldsymbol{A}$) $= n$, the Moore-Penrose inverse can be defined as*

$$\boldsymbol{A}^+ = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}$$

.

The following are a list of property of some of the matrix operations.

- $(\boldsymbol{AB})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1}$
- $(\boldsymbol{A}^T)^{-1} = (\boldsymbol{A}^{-1})^T$
- $(\boldsymbol{AB})^T = \boldsymbol{B}^T\boldsymbol{A}^T$
- $\mathrm{Tr}(\boldsymbol{A}) = \sum_i A_{i,i}$
- $\mathrm{Tr}(\boldsymbol{A}) = \sum_i \mathrm{eig}(\boldsymbol{A})$
- $\mathrm{Tr}(\boldsymbol{AB})$ =$\mathrm{Tr}(\boldsymbol{BA})$
- $\boldsymbol{a}\boldsymbol{a}^T$ =$\mathrm{Tr}(\boldsymbol{a}^T\boldsymbol{a})$

### 3.1.2 Normed Spaces

**Definition 3.13** (Normed Vector Space)**.** *A normed vector space is a vector space that has a norm function that maps the vector to a real, non-negative value. The norm of vector $\boldsymbol{v}$ is typically denoted as $\|\boldsymbol{v}\|$ and satisfies the following axioms for all vector $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{V}$:*

- *Positivity:* $\|\boldsymbol{u}\| \geq 0$
- *Definiteness:* $\|\boldsymbol{u}\| = 0 \implies \boldsymbol{u} = \boldsymbol{0}$

- *Sub-additivity (Triangle Inequality):* $\|\boldsymbol{u}+\boldsymbol{v}\| \leq \|\boldsymbol{u}\| + \|\boldsymbol{v}\|$
- *Positive Homogeneity:* $\|a\boldsymbol{u}\| = |a|\|\boldsymbol{u}\|$

**Definition 3.14** ($\ell_p$ norm)**.** *A commonly and extensively used norm in this thesis is the $\ell_p$ norm for $p \in [1,\infty)$. Formally, the $\ell_p$ norm of $\boldsymbol{u} \in \mathbb{R}^d$ is given as $\|\boldsymbol{u}\|_p = \left(\sum_{i=1}^d |u_i|\right)^{1/p}$. We also use the $\ell_\infty$ norm, also known as the supremum norm, defined as $\|\boldsymbol{u}\|_\infty = \max_i |u_i|$.*

Observe that the $\ell_p$ norms of $\boldsymbol{u} \in \mathbb{R}^d$ satisfies these following inequalities:

- $\|\boldsymbol{u}\|_\infty \leq \|\boldsymbol{u}\|_2 \leq \|\boldsymbol{u}\|_1$
- $\|\boldsymbol{u}\|_1 \leq \sqrt{d}\|\boldsymbol{u}\|_2 \leq d\|\boldsymbol{u}\|_\infty$
- In general, for all $1 \leq p < r : \|\boldsymbol{u}\|_r \leq \|\boldsymbol{u}\|_p \leq d^{1/p-1/r}\|\boldsymbol{u}\|_r$

Note that these results hold in general in all normed vector spaces. The $\ell_2$ norm, also known as the Euclidean norm, would be used extensively in our analysis and the proofs of our theorem. We want to note here that $\ell_2$ norms are rotational-invariant and that $\ell_2$ norms are related to the inner product such that $\langle \boldsymbol{u}, \boldsymbol{u} \rangle = \|\boldsymbol{u}\|_2^2$. As per described by the geometric interpretation of inner product $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$ is equivalent to $\|\boldsymbol{u}\|_2\|\boldsymbol{v}\|_2 \cos\theta$.

As in the case of vector spaces, a norm function of a matrix is a function that maps the matrix to a real, non-negative value. The norm of matrix $\boldsymbol{A}$ is typically denoted as $\|\boldsymbol{A}\|$ and satisfies the following axioms. For all matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{M}_{m\times n}$:

- Positivity: $\|\boldsymbol{A}\| \geq 0$
- Definiteness: $\|\boldsymbol{A}\| = 0 \implies \boldsymbol{A} = \boldsymbol{0}_{m\times n}$
- Sub-additivity (Triangle Inequality): $\|\boldsymbol{A}+\boldsymbol{B}\| \leq \|\boldsymbol{A}\| + \|\boldsymbol{B}\|$
- Positive Homogeneity: $\|a\boldsymbol{A}\| = |a|\|\boldsymbol{A}\|$

Additionally, a matrix norm is called an induced norm if the matrix norm is induced by a vector norm by the following

$\|\boldsymbol{A}\| = \sup\{\|\boldsymbol{A}\boldsymbol{x}\| : \boldsymbol{x} \in \mathbb{R}^n \text{ with } \|\boldsymbol{x}\| = 1\}$. In addition to the axioms of matrix norms, induced norms are also sub-multiplicative

- $\|\boldsymbol{AB}\| < \|\boldsymbol{A}\|\|\boldsymbol{B}\|$
- $\|\boldsymbol{A}\boldsymbol{x}\| \leq \|\boldsymbol{A}\|\|\boldsymbol{x}\|$

**Definition 3.15** (sub-Gaussian norm (Vershynin, 2018))**.** *The sub-Gaussian norm $\|X\|_{\psi_2}$ of a random variable $X$ is the smallest value of $K_4$ such that $E[\exp(X^2/K_4^2)] \leq 2$. Vershynin (2018) also noted that the following parameters $K_i > 0$ in the property below differs from each other by at most a constant factor.*

1. *The tails of $X$ satisfy*

$$Pr\{|X| > t\} \leq 2\exp(-t^2/K_1^2) \text{ for all } t \geq 0$$

2. *The moments of $X$ satisfies*

$$\|X\|_P = (E[|X|^P])^{1/P} \leq K_2\sqrt{P} \text{ for all } P \geq 0$$

3. *The Moments Generating Function of $X^2$ satisfies,*

$$E[\exp(\lambda^2 X^2)] \leq \exp(K_3^2\lambda^2)$$

4. *The Moments Generating Function of $X^2$ is bounded at some point, namely $E[\exp(X^2/K_4^2)] \leq 2$*

*Moreover, any random variable satisfying any of the properties 1–4 is sub-Gaussian.*

## 3.2 Useful inequalities

The following are some inequalities that are useful to derive the results of our theorems.

### 3.2.1 Means

Define the $p$-th power mean of a finite set of positive numbers $S$ to be

$$PM(S,p) = \sqrt[p]{\sum_{s\in S} \frac{s^p}{|S|}}$$

- The arithmetic mean $AM(S) = PM(S,1)$
- The harmonic mean $HM(S) = PM(S,-1)$
- The geometric mean $GM(S) = \lim_{p\to 0} PM(S,p) = \sqrt[|S|]{\prod_{s\in S} s}$
- The maximum $\max(S) = \lim_{p\to\infty} PM(S,p)$
- The minimum $\min(S) = \lim_{p\to -\infty} PM(S,p)$
- $PM(S,p_0) \leq PM(S,p_1)$ for $p_0 \leq p_1$
- $HM(S) \leq GM(S) \leq AM(S)$

### 3.2.2 Expectations and Variances

Let $X$ and $Y$ be random variables. If an inequality includes a function $f$ of a random variable $X$, assume that the expectation $\mathrm{E}[f(X)]$ exists.

- If $g(X) \leq h(X)$, then $\mathrm{E}[g(X)] \leq \mathrm{E}[h(X)]$
- **(Holder)** If $p, q$ satisfy $\frac{1}{p} + \frac{1}{q} = 1$, then $\mathrm{E}[XY] \leq (\mathrm{E}[X^p])^{\frac{1}{p}}(\mathrm{E}[X^q])^{\frac{1}{q}}$
- **(Holder)** For $p > 1$, $\mathrm{E}[X] \leq \sqrt[p]{\mathrm{E}[X]^p}$
- **(Jensen)** For a convex function $g$, If $X \geq Y$, then $\mathrm{E}[g(X)] \geq g(\mathrm{E}[X])$
- **(Cauchy-Schwartz)** $\mathrm{E}[|XY|] \leq \sqrt{\mathrm{E}[X^2]\mathrm{E}[Y^2]}$
- **(Liapounov)** For $s \geq r \geq 1$, $\sqrt[r]{\mathrm{E}[X^r]} \leq \sqrt[s]{\mathrm{E}[X^s]}$
- **(Minkowski)** For $p \geq 1$, $\sqrt[p]{(\mathrm{E}[X] + \mathrm{E}[Y])^p} \leq \sqrt[p]{\mathrm{E}[X^p]} + \sqrt[p]{\mathrm{E}[Y^p]}$
- $e^x \geq (1 + \frac{x}{n})^n \geq 1 + x$, for $n > 1, |x| \leq n$
- $\frac{x}{1+x} \leq \ln(1+x)$

- $1 - x \leq \frac{1}{1+x} \leq 1 - x + x^2$ for $0 \leq x < 1$

- $\ln x \leq \sqrt{x}$

## 3.3 Concentration of Measures

The concentration of measure, sometimes referred to as tail inequalities and concentration inequalities, gives a probability bound for the deviation of a random variable to a fixed value, typically to the expected value of the random variable. Concentration of measures has been a topic of interest in probabilistic analysis. Moreover, high dimensional datasets are found to have structures benefiting from results from concentration of measures (sometimes referred to as "Blessing of Dimensionality"). In this section, we will introduce known results from concentration measures. These results are taken from various textbooks, including Concentration Inequalities (Boucheron et al., 2013) and High Dimensional Probabilities (Vershynin, 2018)

**Lemma 3.1** (Markov's inequality)**.** *Let $X$ be a non-negative random variable with $\mathbb{E}\left[X\right] < \infty$, then $Pr\{X > t\} \leq \frac{\mathbb{E}[X]}{t}$.*

The proof of this lemma is folk lore

*Proof.* Let $t > 0$, define

$$Y := \begin{cases} 0, & \text{if } X \leq t; \\ t, & \text{if } X > t; \end{cases}$$

Observe that $Y \leq X$, therefore

$$\mathbb{E}\left[Y\right] \leq \mathbb{E}\left[X\right]$$

$$\mathbb{E}\left[Y\right] = t\Pr\left\{X > t\right\} \leq \mathbb{E}\left[X\right]$$
$$\Pr\left\{X > t\right\} \leq \frac{\mathbb{E}\left[X\right]}{t}$$

□

Markov's inequality holds for any non-decreasing, integrable functions of $X$. One convenient trick to obtain tighter bounds is to transform $Y := |X - \mathbb{E}\left[X\right]|$

and apply Markov's inequality on $Y^2$ giving us what is commonly known as Chebychev's inequality.

**Lemma 3.2** (Chebychev's inequality)**.** *Let* $\mathrm{x}$ *be a random variable with* $\mathbb{E}[\mathrm{x}^2] < \infty$*, then* $Pr\{|\mathrm{x} - \mathbb{E}[\mathrm{x}]| > t\} \leq \frac{Var[\mathrm{x}]}{t^2}$.

*Proof.* We let $\mathrm{y} = |\mathrm{x} - \mathbb{E}[\mathrm{x}]|^2$ and observe that $\mathbb{E}[\mathrm{y}] = \mathrm{Var}[\mathrm{x}]$. We then apply Markov's Inequality on y and obtain

$$\Pr\{|\mathrm{x} - \mathbb{E}[\mathrm{x}]| > t\} = \Pr\left\{|\mathrm{x} - \mathbb{E}[\mathrm{x}]|^2 > t^2\right\} = \Pr\left\{\mathrm{y} > t^2\right\} \leq \frac{\mathrm{Var}[\mathrm{x}]}{t^2}$$

. □

Chebychev's inequality is historically one of the more prominent result in concentration inequalities. This is primarily due to the second moments (variance) being easy to handle, is intuitive explainable, and well-studied for many of the random distributions. One useful extension of Chebychev's inequality is the "one-Sided Chebychev's inequality" also known as Cantelli's inequality. This inequality was obtained by Cantelli, by showing that the probability bounds hold for an arbitrary choice of $\lambda$, and then choosing a value of $\lambda$ that minimises the inequality. This technique is a commonly used trick to obtain tighter bounds on the concentration inequalities.

**Lemma 3.3** (Cantelli's inequality)**.** *Let* $\mathrm{x}$ *be a random variable with* $\mathbb{E}[\mathrm{x}^2] < \infty$*, then for* $t > 0$ $Pr\{\mathrm{x} - \mathbb{E}[\mathrm{x}] > t\} \leq \frac{Var[\mathrm{x}]}{Var[\mathrm{x}]+t^2}$.

*Proof.* We let $\mathrm{y} = \mathrm{x} - \mathbb{E}[\mathrm{x}]$ and observe that $\mathbb{E}[\mathrm{y}] = 0$ and $\mathbb{E}[\mathrm{y}^2] = \mathrm{Var}[\mathrm{x}]$

$$\Pr\{\mathrm{x} - \mathbb{E}[\mathrm{x}] > t\} = \Pr\{\mathrm{y} > t\} = \Pr\{\mathrm{y} + \lambda > t + \lambda\} < \frac{Var(\mathrm{x}) + \lambda^2}{(t+\lambda)^2}$$

Differentiating the last inequality shows that the $\lambda^*$ that minimizes the inequality is $\lambda^* = \dfrac{\mathrm{Var}[\mathrm{x}]}{t}$. Substituting $\lambda^*$ into the inequality gives us

$$\Pr\{\mathrm{x} - \mathbb{E}[\mathrm{x}] > t\} \leq \frac{\mathrm{Var}[\mathrm{x}]}{\mathrm{Var}[\mathrm{x}] + t^2}$$

□

The techniques used to derive Chebychev's inequalities can be extended to the higher moments. One such method is to apply Markov's inequality to the Moment Generating Function (Laplace Transform of the Random Variable). This is what is known as the Cramer-Chernoff's method and often give a much tighter probability bound. This improvement in probability bounds comes from the Chernoff's bound giving an exponential decay while the Chebychev's inequality implying a s an inverse polynomial decay.

**Lemma 3.4** (Chernoff's Bound)**.** *Let* $\mathrm{x}$ *be a random variable with* $\mathbb{E}\left[\mathrm{x}\right] < \infty$*, then* $Pr\{\mathrm{x} - \mathbb{E}\left[\mathrm{x}\right] > \lambda t\} \leq \dfrac{\mathbb{E}\left[\exp \lambda \mathrm{x}\right]}{\exp \lambda t}$.

One useful application of Chernoff's Bound is Hoeffding's Bound which gives an exponentially decaying bound on the sum of random variables.

**Lemma 3.5** (Hoeffding, 1963 (Hoeffding, 1963))**.** *Let* $X_1, X_2, \ldots, X_k$ *be independent random variables such that,* $\forall i \in 1, 2, \ldots, k$ *we have* $X_i - E[X_i] \in [a_i, b_i]$ *with probability 1. Denote by* $S_k := \sum_{i=1}^{k}(X_i - E[X_i])$ *and fix* $t > 0$*. Then:*

$$Pr\{|S_k| \geq t\} \leq 2\exp\left(-\frac{2t^2}{\sum_{i=1}^{k}(b_i - a_i)^2}\right)$$

*Proof.* Without loss of generality let $\mathrm{E}[X_i] = 0$, or else we let $X_i = X_i - \mathrm{E}[X_i]$. Observe that $S_k := \sum_{i=1}^{k} X_i$. Observe that, by independence, $\mathrm{E}[\exp(\lambda S_k)] = \prod_{i=1}^{k} \mathrm{E}[\exp(\lambda X_i)]$. Note that exponentials are a convex function and for $a \leq X_i \leq b$, and $\lambda > 0$,

$$\exp(\lambda X_i) \leq \frac{b - X_i}{b - a} e^{sa} + \frac{X_i - a}{b - a} e^{sb}$$

Applying expectation to both side of the inequality gives us

$$\begin{aligned}\mathrm{E}[\exp(\lambda X_i)] &\leq \frac{b - \mathrm{E}[X_i]}{b - a} e^{sa} + \frac{\mathrm{E}[X_i] - a}{b - a} e^{sb} \\ &= \frac{b}{b - a} e^{\lambda a} - \frac{a}{b - a} e^{\lambda b}\end{aligned}$$

Let $\theta = -\frac{a}{b-a} > 0$.

$$\begin{aligned}\mathrm{E}[\exp(\lambda X_i)] \leq (1 - \theta)e^{\lambda b} + \theta e^{\lambda a} &= e^{\lambda a}\left(1 - \theta + \theta e^{\lambda(b-a)}\right) \\ &= e^{-\lambda\theta(b-a)}\left(1 - \theta + \theta e^{\lambda(b-a)}\right)\end{aligned}$$

Let $\psi(u) := -\theta u + \log\left(1-\theta+\theta e^{u}\right)$ where $u = \lambda(b-a)$.

Taylor's theorem states that for every $u$ there exists a $v$ between 0 and $u$ such that $\psi(u) = \psi(0) + u\psi'(0) + \frac{1}{2}u^2\psi''(v)$.

Note that $\psi(0) = 0$ and $\psi'(0) = \theta + \frac{\theta e^u}{1-\theta+\theta e^u}|_{u=0} = 0$

$$\psi''(v) = \frac{\theta e^v\left(1-\theta+\theta e^v\right) - \theta^2 e^{2v}}{(1-\theta+\theta e^v)^2} = \frac{\theta e^v}{1-\theta+\theta e^v}\left(1 - \frac{\theta e^v}{1-\theta+\theta e^v}\right)$$

. Observe that $\dfrac{\theta e^v}{1-\theta+\theta e^v}$ is bounded between $0 \leq \dfrac{\theta e^v}{1-\theta+\theta e^v} \leq 1$ and therefore $0 \leq \psi''(v) \leq 1/4$.

Therefore, $\psi(u) \leq \frac{1}{2}u^2(\frac{1}{4}) = \frac{1}{8}\lambda^2\left(b-a\right)^2$. Replacing this equation into the Chernoff bound and applying gives us Hoeffding's inequality.

$$\Pr\left\{S_k - \mathrm{E}[S_k] \geq t\right\} \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^{k}(b_i-a_i)^2}\right)$$

and applying symmetry gives us the two sided bound

$$\Pr\left\{|S_k - \mathrm{E}[S_k]| \geq t\right\} \leq 2\exp\left(-\frac{2t^2}{\sum_{i=1}^{k}(b_i-a_i)^2}\right)$$

□

Hoeffding inequality implies a sub-Gaussian tail inequality for the sum $S_k$ based on the range of $X_i$. However, in the cases where the variance of the sum $S_k$ is much smaller than $\sum_{i=1}^{k}(b_i-a_i)^2$ we can obtain a significantly tighter tail-bound of the sum $S_k$ using Bernstein-Bennett's Inequality.

**Lemma 3.6** (General Hoeffding's inequality, (Theorem 2.6.3 Vershynin (2018))). *Let $X_1, \ldots, X_n$ be independent mean zero, sub-Gaussian random variables, and $\boldsymbol{a} = (a_1, \ldots a_n) \in \mathbb{R}^n$. Then for every $t \geq 0$, we have*

$$Pr\left\{\left|\sum_{i=1}^{N} a_i X_i\right| > t\right\} \leq 2\exp(-\frac{ct^2}{K^2\|\boldsymbol{a}\|_2^2}) \text{ where } K = \max_i \|X_i\|_{\psi_2}$$

**Lemma 3.7** (Bennett Bound, 1962). *(Bennett, 1962)] Let $X_1, X_2, \ldots, X_k$ be independent random variables with finite variance. Also, assume $\sum_{i=1}|X_i| \leq a$. Denote by $S_k := \sum_{i=1}^{k} X_i - E[X_i]$ and $V_k := E\left[\sum_{i=1}^{k}(X_i - E[X_i])^2\right]$ and fix $t > 0$. Then:*

$$Pr\{|S_k| \geq t\} \leq 2\exp\left(-V_k\phi\left(\frac{t}{aV_k}\right)\right)$$

*where $\phi(x) = (1+x)\log(1+x) - x$.*

**Corollary 3.8** (Bernstein-Bennett Bound). *Using the same conditions as stated in Lemma 3.7,*

$$Pr\{|S_k| \geq t\} \leq 2\exp\left(\frac{-t^2}{V_k + \frac{1}{3}at}\right)$$

*Proof.* Without loss of generality let $\mathrm{E}[X_i] = 0$, or else we let $X_i = X_i - \mathrm{E}[X_i]$. Observe that $S_k := \sum_{i=1}^{k} X_i$. Observe that, $\mathrm{E}\left[\exp\left(\lambda S_k\right)\right] = \mathrm{E}\left[\exp\left(\lambda \sum_{i=1}^{k} X_i\right)\right]$. Expanding the Taylor series for exp gives us

$$\mathrm{E}\left[\exp\left(\lambda\sum_{i=1}^{d} X_i\right)\right] = \mathrm{E}\left[\sum_{n=0}^{\infty}\left(\frac{\lambda^n}{n!}\left(\sum_{i=1}^{d} X_i\right)^n\right)\right]$$

Expanding the first 3 terms of $n$ gives us

$$\mathrm{E}\left[\exp\left(\lambda\sum_{i=1}^{d} X_i\right)\right] = \mathrm{E}\left[1 + \lambda\sum_{i=1}^{k}\mathrm{x}_i + \frac{\lambda^2}{2!}\left(\sum_{i=1}^{k}\mathrm{x}_i\right)^2 + \sum_{n=3}^{\infty}\frac{\lambda^n}{n!}\left(\sum_{i=1}^{k} X_i\right)^2\left(\sum_{i=1}^{k} X_i\right)^{n-2}\right]$$

Note that since $\mathrm{E}[x_i] = 0$ and $X_i$ is independent, this implies that $\mathrm{E}\left[(\sum_{i=1}^{k} X_i)^2\right] = \mathrm{E}\left[\sum_{i=1}^{k} X_i^2\right] = V_k$, also observe that $\left(\sum_{i=1}^{k} X_i\right)^{n-2} \leq a^{n-2}$.

Applying linearity of expectations, we have

$$\begin{aligned}
=&1 + \lambda\mathrm{E}\left[\sum_{i=1}^{k}\mathrm{x}_i\right] + \frac{\lambda^2}{2}\mathrm{E}\left[\sum_{i=1}^{k} X_i^2\right] + \sum_{n=3}^{\infty}\frac{\lambda^n}{n!}\mathrm{E}\left[\left(\sum_{i=1}^{k} X_i\right)^2\left(\sum_{i=1}^{k} X_i\right)^{n-2}\right]\\
=&1 + 0 + \frac{\lambda^2}{2}V_k + \sum_{n=3}^{\infty}\frac{\lambda^n}{n!}\mathrm{E}\left[\left(\sum_{i=1}^{k} X_i\right)^2\left(\sum_{i=1}^{k} X_i\right)^{n-2}\right]\\
\leq&1 + \frac{\lambda^2}{2}V_k + \sum_{n=3}^{\infty}\frac{\lambda^n}{n!}V_k a^{n-2}
\end{aligned}$$

Note that the last inequality can also be written as $1 + V_k((\exp(\lambda a) - 1 - \lambda a)$ and noting that $1 + x \leq \exp x$. We have

$$\mathrm{E}\left[\exp\left(\lambda S_k\right)\right] \leq \exp\left(V_k\left(\exp\lambda a - 1 - \lambda a\right)\right)$$

Applying Chernoff's Bound, we have

$$\Pr\left\{S_k > t\right\} \leq \frac{\exp\left(V_k\left(\exp\lambda a - 1 - \lambda a\right)\right)}{\exp\lambda t}$$

Rearranging, we have

$$\Pr\left\{S_k > t\right\} \leq \exp\left(-\lambda t + V_k\left(\exp\lambda a - 1 - \lambda a\right)\right)$$

Optimizing w.r.t. $\lambda$ and choosing $\lambda = \frac{1}{a}\log\left(1 + \frac{t}{aV_k}\right)$ gives us

$$\Pr\left\{S_k > t\right\} \leq exp\left(-\left(\frac{t}{a} + V_k\right)log(1 + \frac{t}{aV_k}) - \frac{t}{a}\right)$$

$$\exp -V_k \left(1+\frac{t}{aV_k}\right) \log\left(1+\frac{t}{aV_k}\right) - \frac{t}{aV_k}$$

To obtain the two sided bound, let $X_i = -X_i$ and use symmetry to obtain the lower bound and to obtain Corollary 3.8, observe that $(1+x)\log(1+x) - x \geq \frac{x^2}{2+\frac{2}{3}x}$ and apply the substitution. □

**Lemma 3.9** (Hoeffding (1963), Theorem 4)**.** *Let $\chi = (x_1, x_2, \ldots, x_d)$ be a finite population of $d > 1$ points, $X_1, \ldots X_k$ denote a random sample without replacement from $\chi$ and $Y_1, \ldots Y_k$ denote a random sample with replacement from $\chi$. If $f : \mathbb{R} \mapsto \mathbb{R}$ is continuous and convex, then $E[f \sum_{i=1}^k X_i] \leq E[f \sum_{i=1}^k Y_i]$*

As noted by Bardenet and Maillard (2015), Lemma 3.9 implies that the concentration results above can be transferred to the setting of sampling without replacement. Moreover, a direct consequence of the lemma is that the tail inequalities for a random sample without replacement would be tighter than what is given by Lemma 3.5 and 3.7.

**Theorem 3.10** (Hoeffding-Serfling Inequality (Bardenet and Maillard, 2015))**.** *Let $\chi = (x_1, x_2, \ldots, x_d)$ be a finite population of $d > 1$ points and $a = \min(\chi)$ and $b = \max(\chi)$. Let $(X_i, \ldots, X_k)$ be a list of size $k < d$ sampled without replacement from $\chi$. Let $S_n = \sum_{i=1}^k X_i$. Then, for all $\epsilon > 0$ the following concentration bound holds:*

$$Pr\{|S_k - E[S_k]| \geq t\} \leq 2\exp\left(-\frac{2t^2}{(1-k/d)((k+1)/k)(b-a)^2}\right)$$

**Theorem 3.11** (Bernstein-Serfling Inequality (Bardenet and Maillard, 2015))**.** *Let $\chi = (x_1, x_2, \ldots, x_d$ be a finite population of $d > 1$ points and $a = \min(\chi)$ and $b = \max(\chi)$. Let $\mu = \frac{1}{d}\sum_{i=1}^d (x_i)$ and $\sigma^2 = \frac{1}{d}\sum_{i=1}^d (x_i - \mu)^2$. Let $(X_i, \ldots, X_k)$ be a list of size $k < d$ sampled without replacement from $\chi$. Let $S_n = \sum_{i=1}^k X_i$, and $\gamma^2 = (1 - \frac{k-1}{d})\sigma^2 + \frac{k-1}{d}(b-a)\sigma\sqrt{\frac{2\log(1/\delta)}{n}}$. Then, for all $\epsilon > 0$ the following concentration bound holds:*

$$Pr\{|S_k - E[S_k]| \geq nt\} \leq 2\exp\left(-\frac{nt^2/2}{\gamma^2 + (2/3)(b-a)t}\right) + \delta$$

## 3.4 Statistical Learning Theory

In this section we will introduce some definitions and theorems for statistical learning, in particular *Probably Approximately Correct* (PAC) learning. These definitions and proofs are taken from Shalev-Shwartz and Ben-David (2014).

**Definition 3.16** (PAC Learnability (Shalev-Shwartz and Ben-David, 2014))**.** *A hypothesis class $\mathcal{H}$ is PAC learnable if there exist a function $\mathcal{M}_{\mathcal{H}} : (0,1)^2 \mapsto \mathbb{N}$ with the following property.*

*For every $\epsilon, \delta \in (0,1)$, and every distribution $\mathcal{D}$ over $\boldsymbol{X}$ and for every labelling function $f : \boldsymbol{X} \mapsto \{0,1\}$, if the realizable assumption holds with respect to $\mathcal{H}, \mathcal{D}$ and $f$, then when running the learning algorithm on $n \geq \mathcal{M}_{\mathcal{H}}(\epsilon, \delta)$ i.i.d. examples generated by $\mathcal{D}$ and labelled by $f$ the algorithm returns a hypothesis $h$ such that with probability at least $1-\delta$ over the choice of examples $L_{\mathcal{D},f}(h) \leq \epsilon$.*

$\mathcal{M}_{\mathcal{H}}$ is known as the sample complexity, and $L_{\mathcal{D},f}(h)$ is known as the true error or risk of the prediction rule $h$. The above definition also implies the following corollary.

**Corollary 3.12.** *Every finite hypothesis class is PAC learnable with sample complexity*

$$\mathcal{M}_{\mathcal{H}}(\epsilon, \delta) \leq \left\lceil \frac{\log(|\mathcal{H}|/\delta)}{\epsilon} \right\rceil$$

Now, definition 3.16 assumes that a labelling function $f$ exists such that the function $f$ can fully determine the label from the features in the data. In practical problems, this assumption may not hold (for instance, there may be two observations in the data that may have different labels even though the feature-values are identical). The following relaxes this realizability assumption and defines what is called Agnostic PAC learnability.

**Definition 3.17** (Agnostic PAC Learnability (Shalev-Shwartz and Ben-David, 2014))**.** *A hypothesis class $\mathcal{H}$ is agnostic PAC learnable if there exist a function $\mathcal{M}_{\mathcal{H}} : (0,1)^2 \mapsto \mathbb{N}$ and a learning algorithm with the following property:*

*For every $\epsilon, \delta \in (0,1)$, and every distribution $\mathcal{D}$ over $\boldsymbol{X} \times \boldsymbol{Y}$, when running the learning algorithm on $n \geq \mathcal{M}_{\mathcal{H}}(\epsilon, \delta)$ i.i.d. examples generated by $\mathcal{D}$, the* *algorithm returns a hypothesis $h$ such that with probability at least $1-\delta$ over the choice of $n$ training examples*

$$L_{\mathcal{D}}(h) \leq \min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + \epsilon$$

Let $\boldsymbol{Z}$ be $\boldsymbol{X} \times \boldsymbol{Y}$ where $\boldsymbol{X}$ is the instance set, and $\boldsymbol{Y}$ is the corresponding label. The *risk function* $\mathcal{L}_{\mathcal{D}}$ defined as the expected loss of a classifier $h \in \mathcal{H}$, with respect to probability distribution $\mathcal{D}$ over $\boldsymbol{Z}$. Mathematically,

$$\mathcal{L}_{\mathcal{D}(h)} := \mathrm{E}_{z \sim \mathcal{D}}[l(h,z)]$$

The *empirical risk* $\mathcal{L}_S$ defined as the expected loss of a classifier $h \in \mathcal{H}$ over a given sample $S = (z_1, z_2, ..., z_n)$,

$$\mathcal{L}_S := \frac{1}{n} \sum_{i=1}^{n} l(h, z_i)]$$

As remarked in chapter 2, in classification tasks, we typically use 0-1 loss for classification.

$$l(h,(x,y)) := \begin{cases} 0 & \text{if } h(x) = y \\ \\ 1 & \text{if } h(x) \neq y \end{cases}$$

**Definition 3.18** ($\epsilon$-representative sample (Shalev-Shwartz and Ben-David, 2014))**.** *A training set $S$ is called $\epsilon$-representative with respect to domain $\boldsymbol{Z}$, hypothesis class $\mathcal{H}$, loss function $l$ and distribution $\mathcal{D}$ if*

$$\forall h \in \mathcal{H}, |\mathcal{L}_S(h) - \mathcal{L}_{\mathcal{D}}(h)| \leq \epsilon$$

**Lemma 3.13** (Uniform convergence (Shalev-Shwartz and Ben-David, 2014))**.** *Let $\mathcal{H}$ be a finite hypothesis class, and let $\boldsymbol{Z}$ be the domain of training samples, and let $l : \mathcal{H} \times \boldsymbol{Z} \mapsto [0,1]$ be a loss function. Then $\mathcal{H}$ has the uniform convergence property with sample complexity*

$$\mathcal{M}_{\mathcal{H}}^{VC}(\epsilon, \delta) \leq \left\lceil \frac{\log(2|\mathcal{H}|/\delta)}{2\epsilon^2} \right\rceil$$

*Proof.* Let $\theta_i = \frac{1}{n}\sum_{i=1}^{n} l(h, z_i)$. Since $h$ is fixed and $z_1, \ldots, z_n$ is sampled i.i.d, it follows that $\theta_1, ..., \theta_n$ is a sequence of i.i.d random variable. Observe that

$\mathrm{E}[\theta_i] = L_{\mathcal{D}}(h)$, let $\mathrm{E}[\theta_i] = \mu$. Also observe that $0 \leq \theta_i \leq 1$. Then by Hoeffding's inequality (4.1) we have

$$Pr|\frac{1}{n}\sum_{i=1}^{n}\theta_i - \mu| > \epsilon \leq 2\exp[-2n\epsilon^2]$$

. Applying union bound on $h \in \mathcal{H}$ classifiers give us

$$\Pr\left\{\sup_{h\in\mathcal{H}} |L_S(h) - L_{\mathcal{D}}(h)| > \epsilon\right\} \leq 2|\mathcal{H}|\exp[-2n\epsilon^2]$$

□

**Definition 3.19** (Restriction of $\mathcal{H}$ to $C$ (Shalev-Shwartz and Ben-David, 2014))**.** *Let $\mathcal{H}$ be a class of functions from $\boldsymbol{X}$ to $\{0,1\}$ and let $C = \{c_1, c_2, ..., c_n\} \subset \boldsymbol{X}$. The restriction of $\mathcal{H}$ to $C$ is the set of functions from $C$ to $\{0,1\}$ that can be derived from $\mathcal{H}$.*

$$\mathcal{H}_C = (h(c_1), \ldots, h(c_n)) : h \in \mathcal{H}$$

**Definition 3.20** (Shattering (Shalev-Shwartz and Ben-David, 2014))**.** *A hypothesis class $\mathcal{H}$ is said to shatter a finite set $C \subset \boldsymbol{X}$ if the restriction of $\mathcal{H}$ to $C$ is the set of all functions from $C$ to $\{0,1\}$. that is $|\mathcal{H}_C| = 2^{|C|}$.*

**Definition 3.21** (VC Dimension (Shalev-Shwartz and Ben-David, 2014))**.** *The VC-dimension of a hypothesis class $\mathcal{H}$, denoted as $VCdim(\mathcal{H})$ is the maximal size of a set $C \subset \boldsymbol{X}$ that can be shattered by $\mathcal{H}$. if $\mathcal{H}$ can shatter sets of arbitrarily large size we say that $\mathcal{H}$ has infinite VC-dimension*

In other words, there exists sets of $VCdim(\mathcal{H})$ points that can be shattered by $\mathcal{H}$ but no sets of $VCdim(\mathcal{H}) + 1$ points can be shattered by $\mathcal{H}$.

**Definition 3.22** (Growth function (Shalev-Shwartz and Ben-David, 2014))**.** *Let $\mathcal{H}$ be a hypothesis class. Then the growth function of $\mathcal{H}$ denoted as $\tau_{\mathcal{H}} : \mathbb{N} \mapsto \mathbb{N}$ is defined as*

$$\tau_{\mathcal{H}}(n) := \max_{C\subset\boldsymbol{X}:|C|=n} |\mathcal{H}_C|$$

*In other words, the growth function $\tau_{\mathcal{H}}(n)$ is the number of different functions from a set $C$ of size $n$ to $\{0,1\}$ that can be obtained by restricting $\mathcal{H}$ to $C$*

**Lemma 3.14** (Sauer-Selah Lemma (Shalev-Shwartz and Ben-David, 2014)). *Let $\mathcal{H}$ be a hypothesis class with a VCdim($\mathcal{H}$) $\leq d < \infty$. for all $n$, $\tau_{\mathcal{H}}(n) \leq \sum_{i=0}^{d} \binom{n}{i}$. In particular is $n > d + 1$ then $\tau_{\mathcal{H}}(n) \leq (en/d)^d$.*

In cases where $d \geq 3$ this can be more compactly written as $\tau_{\mathcal{H}}(n) \leq n^d$.

**Theorem 3.15** (The Fundamental Theorem of Statistical Learning (Shalev-Shwartz and Ben-David, 2014)). *Let $\mathcal{H}$ be a hypothesis class of functions from a domain $\boldsymbol{X}$ to $\{0, 1\}$, and let the loss function be the 0-1 loss. Then the following are equivalent:*

- *$\mathcal{H}$ has the uniform convergence property.*
- *Any ERM rule is a successful agnostic PAC learner for $\mathcal{H}$*
- *$\mathcal{H}$ is agnostic PAC learnable*
- *$\mathcal{H}$ is PAC learnable*
- *Any ERM rule is a successful PAC learner for $\mathcal{H}$*
- *$\mathcal{H}$ has a finite VC-dimension*

**Theorem 3.16** (The Fundamental Theorem of Statistical Learning — Quantitative version (Shalev-Shwartz and Ben-David, 2014)). *Let $\mathcal{H}$ be a hypothesis class of functions from a domain $\boldsymbol{X}$ to $\{0, 1\}$, and let the loss function be the 0-1 loss. Assume VCdim($\mathcal{H}$) $= d < \infty$. Then there are absolute constants $C_1, C_2$ such that*

- *$\mathcal{H}$ has the uniform convergence property with sample complexity*
$$C_1 \frac{d + \log(1/\delta)}{\epsilon^2} \leq \mathcal{M}_{\mathcal{H}}^{VC}(\epsilon, \delta) \leq C_2 \frac{d + \log(1/\delta)}{\epsilon^2}$$
- *$\mathcal{H}$ is agnostic PAC learnable with sample complexity*
$$C_1 \frac{d + \log(1/\delta)}{\epsilon^2} \leq \mathcal{M}_{\mathcal{H}}(\epsilon, \delta) \leq C_2 \frac{d + \log(1/\delta)}{\epsilon^2}$$
- *$\mathcal{H}$ is PAC learnable with sample complexity*
$$C_1 \frac{d + \log(1/\delta)}{\epsilon} \leq \mathcal{M}_{\mathcal{H}}(\epsilon, \delta) \leq C_2 \frac{d + \log(1/\delta)}{\epsilon}$$

## 3.5 Notations

The notations used in this thesis are consistent with the notations established in "Deep Learning" by Goodfellow et al. (2016). The following tables summarises the notations

**Number, Arrays and Sets**

| | |
|---|---|
| $a$ | A scalar (integer or real) |
| $\boldsymbol{a}$ | A vector |
| $\boldsymbol{A}$ | A matrix |
| $\boldsymbol{I}_n$ | Identity matrix with $n$ rows and $n$ columns |
| $\boldsymbol{I}$ | Identity matrix with dimensionality implied by context |
| $\boldsymbol{e}^{(i)}$ | Standard basis vector $[0, \dots, 0, 1, 0, \dots, 0]$ with a 1 at position $i$ |
| $\text{diag}(\boldsymbol{a})$ | A square, diagonal matrix with diagonal entries given by $\boldsymbol{a}$ |
| $\mathbb{R}$ | The set of real numbers |
| $\{0, 1\}$ | The set containing 0 and 1 |
| $\{0, 1, \dots, n\}$ | The set of all integers between 0 and $n$ |
| $[a, b]$ | The real interval including $a$ and $b$ |
| $(a, b]$ | The real interval excluding $a$ but including $b$ |
| $a_i$ | Element $i$ of vector $\boldsymbol{a}$, with indexing starting at 1 |
| $a_{-i}$ | All elements of vector $\boldsymbol{a}$ except for element $i$ |
| $A_{i,j}$ | Element $i, j$ of matrix $\boldsymbol{A}$ |
| $\boldsymbol{A}_{i,:}$ | Row $i$ of matrix $\boldsymbol{A}$ |
| $\boldsymbol{A}_{:,i}$ | Column $i$ of matrix $\boldsymbol{A}$ |
| $\boldsymbol{A}^\top$ | Transpose of matrix $\boldsymbol{A}$ |
| $\boldsymbol{A}^+$ | Moore-Penrose pseudoinverse of $\boldsymbol{A}$ |
| $\boldsymbol{A} \odot \boldsymbol{B}$ | Element-wise (Hadamard) product of $\boldsymbol{A}$ and $\boldsymbol{B}$ |
| $\det(\boldsymbol{A})$ | Determinant of $\boldsymbol{A}$ |
| $\|\|\boldsymbol{x}\|\|_p$ | $L^p$ norm of $\boldsymbol{x}$ |
| $\|\|\boldsymbol{x}\|\|$ | $L^2$ norm of $\boldsymbol{x}$ |

### Information Theory and Functions

| | |
|---|---|
| $P(\mathrm{a})$ | A probability distribution over a variable |
| $\mathrm{a} \sim P$ | Random variable a has distribution $P$ |
| $\mathbb{E}_{\mathrm{x}\sim P}[f(x)]$ or $\mathbb{E}[f(x)]$ | Expectation of $f(x)$ with respect to $P(\mathrm{x})$ |
| $\mathrm{Var}(f(x))$ | Variance of $f(x)$ under $P(\mathrm{x})$ |
| $\mathrm{Cov}(f(x), g(x))$ | Covariance of $f(x)$ and $g(x)$ under $P(\mathrm{x})$ |
| $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ | Gaussian distribution over $\boldsymbol{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ |
| $f : \mathbb{A} \rightarrow \mathbb{B}$ | The function $f$ with domain $\mathbb{A}$ and range $\mathbb{B}$ |
| $f \circ g$ | Composition of the functions $f$ and $g$ |
| $f(\boldsymbol{x}; \boldsymbol{\theta})$ | A function of $\boldsymbol{x}$ parametrized by $\boldsymbol{\theta}$. |
| $\log x$ | Natural logarithm of $x$ |
| $x^+$ | Positive part of $x$, i.e., $\max(0, x)$ |
| $\mathbf{1}_{\mathrm{condition}}$ | is 1 if the condition is true, 0 otherwise |
| $p_{\mathrm{data}}$ | The data generating distribution |
| $\hat{p}_{\mathrm{data}}$ | The empirical distribution defined by the training set |
| $\mathbb{X}$ | A set of training examples |
| $\boldsymbol{x}^{(i)}$ | The $i$-th example (input) from a dataset |
| $y^{(i)}$ or $\boldsymbol{y}^{(i)}$ | The target associated with $\boldsymbol{x}^{(i)}$ |
| $\boldsymbol{X}$ | The $m \times n$ matrix with input example $\boldsymbol{x}^{(i)}$ in row $\boldsymbol{X}_{i,:}$ |

As far as possible, we use the following notation as a shorthand for the following

### Notations

| | |
|---|---|
| $d$ | Dimensionality of the dataset |
| $k$ | Projected dimension, typically $k << d$ |
| $n$ | Number of data samples |
| $N$ | Number of classifiers in the ensemble |
| $m$ | Number of classes in the $m$-class classification task |
| $\boldsymbol{P}$ | Random subspace projection matrix, size is implied by context |
| $\boldsymbol{R}$ | Random Projection matrix, size of the matrix is always $k \times d$ |

# 4

# Norm and Dot Product Preservation Guarantees on Random Subspace Projections

**Summary** Random subspace (RS) is a popular dimensionality reduction approach, widely used for generating diverse classifier ensembles. In this chapter, we show that under suitable data-dependent conditions, RS approximately preserves important structure present in the high dimensional data but in a form of a much lower-dimensional representation. Specifically, we show the data-dependent conditions for a Johnson-Lindenstrauss-type (JLL) guarantee for norm and dot-product preservation for random subspace projections. We also show in section 4.2 how these JLL guarantees for random subspace are related to a notion of "regularity" in the original data.

In section 4.3, we corroborate our findings with empirical results on norm preservation using synthetic and real-world datasets, namely natural image data (Weber, 2006), sparse high-dimensional dataset (Guyon et al., 2004), and audio data (Fonseca et al., 2017).

In section 4.5, we discuss the implications of our theory as developed in section 4.2 and empirically demonstrate how a non-uniform feature sampling scheme can (somewhat) improve the norm preserving properties of a random subspace projection.

In section 4.6, we will apply our results to applications in compressive sensing, in particular, the recovery of sparse signal and propose a method on recovering natural images using RS projections.

## 4.1 Background and Motivation

Randomized dimensionality reduction techniques, such as Random Projection (RP) (Dasgupta and Gupta, 2003; Indyk and Motwani, 1998) and Ho's Random Subspace method (RS) (Ho, 1998) are popular approaches for data compression as part of an analysis workflow. There have been many empirical studies show the utility of both dimensionality reduction techniques for machine learning and data mining tasks in practice (Skurichina and Duin, 2002; Ho, 1995; Li and Zhao, 2009; Lai et al., 2006; Kuncheva et al., 2010; Tao et al., 2006).

A key theoretical motivation for RP behind their use is the Johnson-Lindenstrauss lemma (JLL), with the usual constructive proof of also implying the existence of an algorithm with high-probability geometry preservation guarantees on projected data.

Intuitively, we see that RP distorts the Euclidean geometry of the original data somewhat, but with high probability (overdraws of the random matrix $\boldsymbol{R}$) not too much because of JLL. At the same time, RP allows one to work with a much-compressed representation of the original data. Thus, RP can yield, with the same probability, approximate solutions with performance guarantees for *any* algorithm whose output depends only on the Euclidean geometry of a set of observations. For example, linear classification and regression algorithms, clustering algorithms such as $k$-means, and even non-linear classifiers such as $k$-Nearest Neighbours all fit this bill.

We also note that in the field of compressive sensing the sensing matrix must afford a JLL-type guarantee, also called the Restricted Isometry Property (RIP). A sensing matrix satisfying the RIP when applied to a signal that admits a known sparse representation is the key to successfully reconstructing such a signal from its compressed representation (Baraniuk et al., 2008).[1] However, RP is costly to apply to large or high-dimensional datasets since it requires a matrix-matrix multiplication to implement the projection, and furthermore, the projected features may be hard to interpret, eroding the benefits of working with compressed data. Moreover, as far as we are aware, the only known constructions for $\boldsymbol{R}$ satisfying the JLL comprise sampling the entries from symmetric zero-mean sub-Gaussian distributions.

On the other hand, RS is a particularly appealing approach for dimensionality reduction because it merely involves selecting a subset of data feature indices randomly without replacement, and so does not require a matrix-matrix multiplication to implement the projection and it retains (a subset of) the original features. RS is therefore computationally far more efficient in practice, and more interpretable than RP, but there is little theory to explain its effectiveness and, in particular, there is no known JLL guarantee for RS. Our aim here is to obtain JLL-type guarantees for RS, thus improving our understanding of this approach and at the same time providing a further route to simple, efficient, approximation algorithms with performance guarantees for a broader range of applications.

In all of the following, we assume, without loss of generality, that we possess a (fixed) set of $n$, $d$-dimensional real-valued vector observations to be projected, $\mathcal{T}_N := \{X_i \in \mathbb{R}^d\}_{i=1}^n$ and we may choose an integer, $k$, where $k \in \{1, 2, \ldots, d\}$ as the projection dimension.

## 4.2 Random Subspace as a JLL-like projection

In this section, we present the following four theorems, showing that an RS projection implies a data-dependent JLL-type guarantee. The strength of the

[1] Note that the captured signal need not be sparse, and it generally will not be. However, in order to reconstruct it from its RP form, some basis in which the signal has sparse representation must be known.

provided guarantees depends on how regular the representation in which we are working in, where regularity is measured by (an upper bound on) the squared population coefficient of variation if we consider the elements of a vector as a finite population of size $d$. Our first two theorems are simple Chernoff-Hoeffding type bounds, while the latter two are typically tighter Bernstein type bounds. The second and fourth bounds become much tighter than the first and third as $k \nearrow d$, but they give a similar guarantee to the others when $k \ll d$. Meanwhile, our third and fourth bounds are considerably tighter than the first two when the distribution of feature values is heavy-tailed, e.g. for sparse datasets.

We provide some intuition about the relative performances of our bounds in figure 4.2. Although, as far as we know, these results are novel. As the proofs for our bounds are elementary and use standard tools, we defer them to the Appendix A. For notational and analytical convenience we will write a particular RS projection in the form of a matrix $\boldsymbol{P}$, where $\boldsymbol{P}$ is a $d \times d$ diagonal matrix with all entries zero except for $k$ diagonal entries set to 1 with their indices chosen by simple random sampling without replacement from $\{1, 2, \ldots, d\}$. Note that left-multiplying a $d \times n$ data matrix with $P$ is mathematically equivalent to RS – viewed as a projection of the original data to a subspace of dimension $k$ embedded in $\mathbb{R}^d$ – although in practice this is not how RS is usually implemented. For convenience we also define $\boldsymbol{x}_i^2 := (X_{i1}^2, X_{i2}^2, \ldots, X_{id}^2)^T$ the vector with its entries the squared components of $\boldsymbol{X}^{(i)}$.

**Theorem 4.1** (Basic Chernoff Bound). *Let $\mathcal{T}_N := \{X_i \in \mathbb{R}^d\}_{i=1}^n$ be a set of $n$ points in $\mathbb{R}^d$ satisfying, $\forall i \in \{1, 2, \ldots, n\}$, $\|X_i^2\|_\infty \leq \frac{c}{d}\|X_i\|_2^2$ where $c \in \mathbb{R}_+$ is a constant $1 \leq c \leq d$. Let $\epsilon, \delta \in (0, 1]$, and let $k \geq \frac{c^2}{2\epsilon^2} \ln \frac{n^2}{\delta}$ be an integer. Let $P$ be a random subspace projection from $\mathbb{R}^d \mapsto \mathbb{R}^k$. Then with probability at least $1 - \delta$ over the random draws of $P$ we have, for every $i, j \in \{1, 2, \ldots, n\}$:*

$$(1 - \epsilon)\|X_i - X_j\|_2^2 \leq \frac{d}{k}\|PX_i - PX_j\|_2^2 \leq (1 + \epsilon)\|X_i - X_j\|_2^2$$

**Theorem 4.2** (Chernoff-Serfling Bound)**.** *Let $\mathcal{T}_N := \{X_i \in \mathbb{R}^d\}_{i=1}^n$ be a set of $N$ points in $\mathbb{R}^d$ satisfying, $\forall i \in \{1, 2, \ldots, n\}$, $\|X_i^2\|_\infty \leq \frac{c}{d}\|X_i\|_2^2$ where $c \in \mathbb{R}_+$ is a constant $1 \leq c \leq d$. Let $\epsilon, \delta, f_k \in (0, 1]$, where $f_k := (k-1)/d$ and let $k$ such that $k/(1-f_k) \geq \frac{c^2}{2\epsilon^2} \ln \frac{n^2}{\delta}$ be an integer. Let $P$ be a random subspace projection from $\mathbb{R}^d \mapsto \mathbb{R}^k$. Then with probability at least $1-\delta$ over the random draws of $P$ we have, for every $i, j \in \{1, 2, \ldots, n\}$:*

$$(1-\epsilon)\|X_i - X_j\|_2^2 \leq \frac{d}{k}\|P(X_i - X_j)\|_2^2 \leq (1+\epsilon)\|X_i - X_j\|_2^2$$

**Theorem 4.3** (Bernstein-type Bound)**.** *Let $\mathcal{T}_N := \{X_i \in \mathbb{R}^d\}_{i=1}^n$ be a set of $n$ points in $\mathbb{R}^d$ satisfying, $\forall i \in \{1, 2, \ldots, n\}$, $\|X_i\|_4^2 \leq \sqrt{\frac{c'^2}{8d}}\|X_i\|_2^2$ where $c' \in \mathbb{R}_+$ is a constant $1 \leq c' \leq d$. Let $\epsilon, \delta \in (0, 1]$, and let $k \geq \frac{c'^2}{2\epsilon^2} \ln \frac{N^2}{\delta}$ be an integer. Let $P$ be a random subspace projection from $\mathbb{R}^d \mapsto \mathbb{R}^k$. Then with probability at least $1-\delta$ over the random draws of $P$ we have, for every $i, j \in \{1, 2, \ldots, n\}$:*

$$(1-\epsilon)\|X_i - X_j\|_2^2 \leq \frac{d}{k}\|PX_i - PX_j\|_2^2 \leq (1+\epsilon)\|X_i - X_j\|_2^2$$

**Theorem 4.4** (Bernstein-Serfling Bound)**.** *Let $\mathcal{T}_N := \{X_i \in \mathbb{R}^d\}_{i=1}^n$ be a set of $N$ points in $\mathbb{R}^d$ satisfying, $\forall i \in \{1, 2, \ldots, n\}$, $\|X_i\|_4^2 \leq \sqrt{\frac{c'^2}{8d}}\|X_i\|_2^2$ where $c' \in \mathbb{R}_+$ is a constant $1 \leq c' \leq d$. Let $\epsilon, \delta, f_k \in (0, 1]$, where $f_k := (k-1)/d$ and let $k$ such that $k/(1-f_k) \geq \frac{c'^2}{2\epsilon^2} \ln \frac{n^2}{\delta}$ be an integer. Let $P$ be a random subspace projection from $\mathbb{R}^d \mapsto \mathbb{R}^k$. Then with probability at least $1-\delta$ over the random draws of $P$ we have, for every $i, j \in \{1, 2, \ldots, n\}$:*

$$(1-\epsilon)\|X_i - X_j\|_2^2 \leq \frac{d}{k}\|\boldsymbol{P}(X_i - X_j)\|_2^2 \leq (1+\epsilon)\|X_i - X_j\|_2^2$$

*Comment on Theorem 4.3* The proof of theorem 4.3 shows how $c'$ can be calculated easily for certain data representations, (i.e. sparse binary data), moreover, we will also show that $c'$ is significantly smaller than $c$ in sparse data representations.

Furthermore, we also have:

**Corollary 4.5** (to any of the bounds)**.** *Under the conditions of Theorem 4.1, 4.2, 4.3 or 4.4 respectively, for any $\epsilon, \delta \in (0, 1]$, with probability at least $1-2\delta$*

*over the random draws of $P$ we have:*

$$\left(X_i^T X_j - \epsilon\|X_i\|\|X_j\|\right) \leq \frac{d}{k}(PX_i)^T(PX_j) \leq \left(X_i^T X_j + \epsilon\|X_i\|\|X_j\|\right)$$

*A comment on Corollary 4.5.* For RP matrices with zero-mean sub-Gaussian entries, a $1-\delta$ guarantee for projected dot products is proved in Kabán (2015). The proof technique used there is not directly transferable to RS although we speculate that, for small enough constants $c$ or $c'$ respectively, it could be adapted to RS using some results of Matoušek (2008). We do not pursue this further here.

### 4.2.1 Discussion on the Bounds

Our theorems and their corollaries showed that we have high probability guarantees on Euclidean geometry preservation for sufficiently regular datasets when applying RS, provided that the dimension of the projected subspace, $k$, chosen is large enough. We note that up to constant terms, they provide the same guarantees as we have for the existing JLL for RP; therefore, the bounds are of optimal order for any linear dimensionality reduction scheme (Larsen and Nelson, 2014) and, for a fixed $k$ the RS projection is typically orders of magnitude faster than RP. However, there is a trade-off involved since if $c$ or $c'$ is large ($c' > 4$) the projection dimension required will generally be greater than for RP; indeed for RP, our constants can be replaced by a single-digit constant (either 2 or 8) which only depends on the choice of a Gaussian or sub-Gaussian RP matrix $\boldsymbol{R}$ and not on the data. We note however, that $c$ or $c'$ need not be larger in practice. For instance, when the features are approximately normally distributed, (i.e. $\boldsymbol{X} \sim N(0, \frac{1}{\sqrt{d}}I_d)$, $c'$ is approximately 3, giving us similar JLL guarantees as RP projections.

Appendix B summarizes the estimated $c$ and $c'$ values on other distributions and a strategy to construct synthetic data with for a given value of $c$. Table 4.1 shows some observed values of these constants from image data confirming that the values of $c$ are typically smaller than 8 on dense datasets. On the other hand, Table 4.2 gives some observed values for very sparse binary data from a drug discovery problem are much larger than 8.

Our bounds hold for an RS projection of any set of data vectors meeting the given conditions this may seem somewhat surprising. For example, if we consider a binary vector $\boldsymbol{x}$ with only one non-zero component then it is straightforward to check that under RS with probability $1 - (d - k)/d$ the projected vector is the zero vector, which otherwise would have a norm of 1. In neither case is the squared norm of $\boldsymbol{Px}$ close to its expected value $\frac{k}{d}\|X\|_2^2$ in general. Furthermore, it is easy to verify for any vector with $s < d$ non-zeros that the number of non-zero components sampled by RS has a Hypergeometric$(s, d, k)$ distribution and so if $s \ll d$ this problem remains, and the norms of most projections will be very far from their expected value. However, we note that in such cases the regularity constants $c', c \in [1, d]$ will also be close to $d$ and thus there will only be a non-zero probability guarantee of norm preservation for $k = d$ when, of course, the guarantee holds trivially. Thus for RS, it is not possible to avoid some regularity conditions on the data and to also have non-trivial JLL-type norm-preservation guarantees, and for fixed $\epsilon$ the projection dimension $k$ must sometimes be larger than it would be for RP but this is the price to pay for using RS projection, with its various other benefits over RP. On the other hand, with our view of norm preservation as a special case of estimating a population mean (i.e. that of a population of features), classical results from statistical sampling theory suggest that one could reduce $k$ by using a non-uniform sampling scheme, in particular by using stratified – rather than uniform – sampling of the data feature indices. We discuss our findings using stratified sampling in Section 4.5. Finally, we see from our theorem that it is not sparsity of the data *per se* that causes a problem; rather *it is sparsity in the data representation.* Of course, for many important domains, such as images, even though a very sparse representation of the data is possible, the data are typically captured in a basis in which they have a dense representation.

### 4.2.2 Implications for Classification

In this section, we apply the results of our theorems to linear classification and extend the approach of Arriaga and Vempala (1999) on $l$-robust half-spaces to classification of RS-projected data in the presence of a margin. A half-space is said to be $l$-robust, if there is a probability of zero that any point is within an Euclidean distance of $l$ of the boundary of a linear threshold function separating the class supports. Figure 4.1 illustrate an example of a $l$-robust half-space. A key implication of this result is the $(\epsilon, \delta)$-learnability of a RS-projected robust half-space, that is, with probability $1 - \delta$, a hypothesis that is consistent with at least $1 - \epsilon$ of the data distribution is produced by a learning algorithm. We derive the following Theorem 4.6 using a similar proof technique to Arriaga and Vempala (1999).

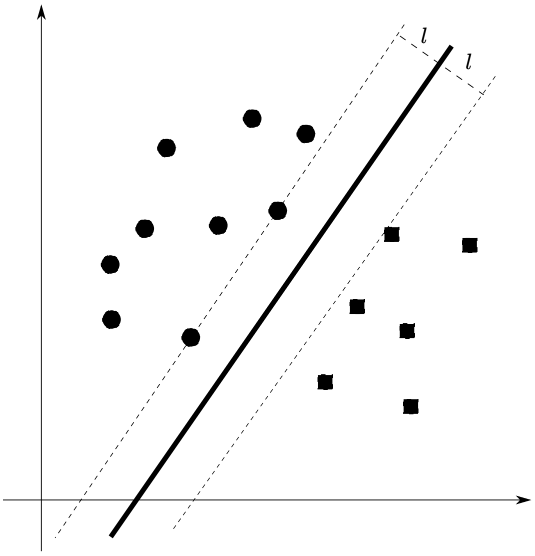

**Figure 4.1:** *Example of an $l$-robust half-space with margins at least $l$.*

**Theorem 4.6.** *An $l$-robust half-space in $\mathbb{R}^d$ can be $(\epsilon, \delta)$-learned by projecting a set of $n$ examples using RS projection to $\mathbb{R}^k$ where*

$$k = \frac{32c'^2}{l^2} \ln \frac{8c'}{\epsilon l \delta}, \quad \textit{and } n = \frac{8k}{\epsilon} \ln \frac{48}{\epsilon} + \frac{4}{\epsilon} \ln \frac{4}{\epsilon}$$

*Proof.* For each $\boldsymbol{x} \in \mathbb{R}^d$, let $\boldsymbol{x}' = \boldsymbol{P}_k \boldsymbol{x}$ be the RS projection of $\boldsymbol{x}$ on to $\mathbb{R}^k$ by selecting $k$ feature values selected uniformly at random without replacement. Define $\boldsymbol{h}$ as a dense normal vector with regularity constant at most $c'$ to the target half-space and $\boldsymbol{h}' = \boldsymbol{P}_k\boldsymbol{h}$ be the projection of $\boldsymbol{h}$. Without loss of generality, we may take $\|\boldsymbol{x}\| = 1, \forall\boldsymbol{x}$ and $\|\boldsymbol{h}\| = 1$ otherwise we can replace $\boldsymbol{x}$ by $\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}$ everywhere. To prove Theorem 4.6, we will require the following events to occur:

- For every $\boldsymbol{x}$, $\|\boldsymbol{P}\boldsymbol{x}\| \leq 1 + \frac{l}{2}$
- $\|\boldsymbol{P}\boldsymbol{h}\| \leq 1 + \frac{l}{2}$
- For every $\boldsymbol{x}$, if $\boldsymbol{h}^T\boldsymbol{x} \geq l$, then $(\boldsymbol{P}\boldsymbol{h})^T(\boldsymbol{P}\boldsymbol{x}) \geq \frac{l}{2}$; else if $\boldsymbol{h}^T\boldsymbol{x} \leq -l$, then $(\boldsymbol{P}\boldsymbol{h})^T(\boldsymbol{P}\boldsymbol{x}) \leq -\frac{l}{2}$

and we will upper bound the "failure probability" of the complementary events for a randomly selected $\boldsymbol{P}$ to prove the theorem. Applying Theorem 4.3 for a single example $\boldsymbol{x}$, and setting $\epsilon = \frac{l}{2}$, the probability that $\|\boldsymbol{P}\boldsymbol{x}\| > 1 + \frac{l}{2}$ is at most

$$\begin{aligned}
2\exp\left(-\frac{kl^2}{16c'^2}\right) &= 2\exp\left(-4\ln\frac{8c'}{\epsilon l\delta}\right)\\
&= 2\left(\frac{\epsilon l\delta}{8c'}\right)^4\\
&< \frac{\delta\epsilon l^2}{4\left(64c'^2\sqrt{\frac{16c'}{\epsilon l\delta}}\sqrt{\frac{48}{\epsilon}}+\sqrt{\frac{4}{\epsilon}}+1\right)}\\
&< \frac{\delta\epsilon l^2}{4(64c'^2\ln\frac{16c'}{\epsilon l\delta}\ln\frac{48}{\epsilon}+\ln\frac{4}{\epsilon}+1)}\\
&= \frac{\delta}{4(n+1)}
\end{aligned}$$

Since $\boldsymbol{h}$ has regularity constant less than $c'$, the failure probability for second event is also at most $\frac{\delta}{4(n+1)}$. Now, union bounding the failure probability for the $n$ examples, and the normal vector $\boldsymbol{h}$, gives us the failure probability of at most $\frac{\delta}{4}$ for the first two events.

By Corollary 4.5, the failure probability for the third event is at most $\frac{\delta}{4(n+1)} < \frac{\delta}{4n}$. Union bounding the failure probability for the $n$ examples gives us a failure probability of at most $\frac{\delta}{4}$.

Finally, applying union bound to the two results above, shows that with probability at least $1 - \delta/2$, all three events hold simultaneously. Observe that the margins in the projected space is at least $\frac{l/2}{1+l/2} > \frac{l}{3}$ with probability at least $1 - \delta/2$. Now, we can apply any standard PAC$(\epsilon, \delta/2)$ learning bound for data with margin of at least $l/3$ (e.g. Freund and Schapire (1999)) □

As Arriaga and Vempala (1999) noted, the implications of this result show that the half-space in $\mathbb{R}^k$ defined by $\boldsymbol{Ph}$ would correctly classify all $n$ examples after a random subspace projection from $\mathbb{R}^d \mapsto \mathbb{R}^k$, with probability at least $1 - \delta/2$ and the generalization error of this classifier would be bounded above by $\epsilon$ with probability $1 - \delta$. Moreover, the margins remain at least $l/3$ after an RS projection. For example, by the results of Minsky and Papert (1969), a perceptron classifier with this generalization error can be learned in at most $9/l^2$ passes over the data.

## 4.3 Empirical Corroboration of Theory

In this section, we present our experimental results, which corroborate our theory developed in Section 4.2.

### 4.3.1 Synthetic Data

We generated synthetic data (random binary strings) using various values of $c$ to control the sparsity, i.e. setting $c \propto d/s$ where $s$ is the number of non-zero entries. We fixed the embedding dimension at $d = 100,000$ and generated $n = 10,000$ instances of data for each value of $c$ and $\epsilon = \{0.05, 0.1, 0.2, 0.5, 1\}$. We calculated the proportion of projected data points whose norm was distorted by more than $\epsilon$ and then compare the upper bounds obtained by Theorems 4.1, and 4.3 against the sample proportion of $\left|\frac{d}{k}\left\|\boldsymbol{Px}\right\|_2^2 - \left\|\boldsymbol{x}\right\|_2^2\right| > \epsilon\left\|\boldsymbol{x}\right\|_2^2$. The results are plotted as in Figure 4.2, where the horizontal axis is $k$, the projected dimension, given in log-scale and the vertical axis is $\delta$ or the proportion of norms violating this inequality.

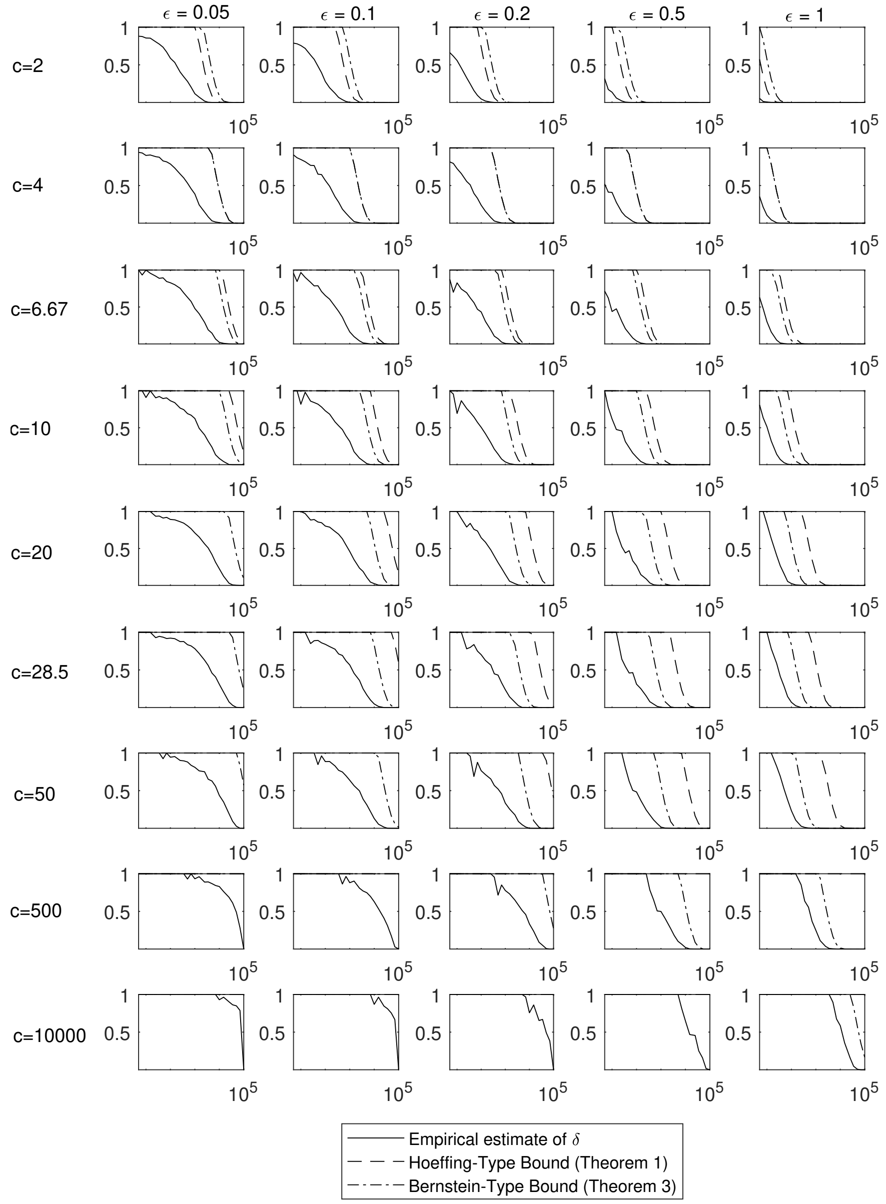

**Figure 4.2:** *Probability bounds of Theorems 4.1 and Theorem 4.3 vs* $Pr\left\{\left|\frac{d}{k}\|PX\|_2^2 - \|X\|_2^2\right| > \epsilon\|X\|_2^2\right\}$ *estimated from 10,000 instances. We controlled c by increasing the sparsity of the data and $\delta$ by increasing the projection dimension k for five fixed values of $\epsilon$. Note that the x-axis is in log scale. Observe that the empirical estimates of $\delta$ is bounded by our theorems.*

We see that our bounds are always upper bounds on the empirical estimates. For low values of $c$ or $c'$ ($c' < 4$), Theorem 4.1 is generally tighter than Theorem 4.3, while for larger values of $c$ or $c'$ this situation is reversed. Similar outcomes with the same general behaviour were obtained for the tighter bounds of Theorems 4.2 and 4.4 – we omit them here. We would like to note that the results hold regardless of the data generator and does not require the features in the data or the sampling scheme to be independent. For a fixed $k$ and $\epsilon$, the proportions $\delta$ norms violating the $\epsilon$-approximate isometry of the projected vector norms depends only on the regularity constant.

### 4.3.2 Real-world Data

Next, when we compare RS projection with two RP variants as well as to principal components analysis (PCA), we see that in practice – given a suitable choice of $k$ – RS works as well as the RP alternatives and is competitive with PCA.

We used three non-synthetic datasets, the first being a collection of natural images (Weber, 2006) similar to those used in the experimental study of Bingham and Mannila (2001) and the second is the DOROTHEA dataset from the 2003 NIPS feature selection challenge (Guyon et al., 2004). The latter is a very sparse and very high-dimensional binary drug-discovery dataset that was split into three for purposes of the NIPS competition. The third dataset is three audio files obtained from freesound.org (Fonseca et al., 2017) and released under creative commons license. These audio files represent the range of everyday sounds namely, a piece of classical music, animal sounds in a forest, and human speech; all sampled at 44100 Hz. The characteristics of these datasets are summarized in Tables 4.1, 4.2 and 4.3 respectively.

For the image data, we used all twenty-three publicly available natural grayscale images from the USC-SIPI natural image dataset, and we omitted the synthetic images. A short description and the sizes of the images are given in Table 4.1. We follow the same protocol as Bingham and Mannila (2001);

| Name | Description | Image Size | $c$ | $c'$ |
|---|---|---|---|---|
| 5.1.09 | Moon Surface | 256x256 | 3.63 | 3.03 |
| 5.1.10 | Aerial | 256x256 | 2.82 | 3.34 |
| 5.1.11 | Airplane | 256x256 | 1.40 | 2.94 |
| 5.1.12 | Clock | 256x256 | 1.59 | 3.11 |
| 5.1.14 | Chemical plant | 256x256 | 5.11 | 3.60 |
| 5.2.08 | Couple | 512x512 | 3.88 | 3.30 |
| 5.2.09 | Aerial | 512x512 | 1.90 | 3.00 |
| 5.2.10 | Stream and bridge | 512x512 | 4.08 | 3.79 |
| 5.3.01 | Man | 1024x1024 | 5.77 | 3.91 |
| 5.3.02 | Airport | 1024x1024 | 7.07 | 3.88 |
| boat.512 | Fishing Boat | 512x512 | 3.42 | 3.23 |
| 7.1.01 | Truck | 512x512 | 5.12 | 3.07 |
| 7.1.02 | Airplane | 512x512 | 2.00 | 2.88 |
| 7.1.03 | Tank | 512x512 | 2.72 | 2.99 |
| 7.1.04 | Car and APCs | 512x512 | 3.92 | 3.10 |
| 7.1.05 | Truck and APCs | 512x512 | 4.94 | 3.27 |
| 7.1.06 | Truck and APCs | 512x512 | 6.93 | 3.36 |
| 7.1.07 | Tank | 512x512 | 4.49 | 3.03 |
| 7.1.08 | APC | 512x512 | 2.76 | 2.94 |
| 7.1.09 | Tank | 512x512 | 3.63 | 3.16 |
| 7.1.10 | Car and APCs | 512x512 | 3.00 | 3.03 |
| 7.2.01 | Airplane (U-2) | 1024x1024 | 21.50 | 6.77 |
| elaine.512 | Girl (Elaine) | 512x512 | 2.85 | 3.34 |

**Table 4.1:** *Properties of Natural Image Dataset. c is the regularity constant in the bounds, which here was calculated from each complete image. $c'$ is the corresponding constant using the Bernstein-type bounds.*

| Name | Number of observations | Features with non-zero variance($d$) | $c$ | $c'$ |
|---|---|---|---|---|
| .test | 800 | 91362 | 139.91 | 33.46 |
| .train | 800 | 88119 | 134.12 | 32.76 |
| .valid | 350 | 72113 | 110.10 | 29.68 |

**Table 4.2:** *Properties of DOROTHEA Dataset. c is the regularity constant in the bounds, which here was calculated from each dataset split. $c'$ is the corresponding constant using the Bernstein-type bounds.*

| Name | Description | Duration (s) | c | c' |
|---|---|---|---|---|
| classical music | Violin Solo from Tchaikovsky's "Danse Arabe". | 71.83 | 7.0320 | 6.866 |
| nature sound | Recording of frog sounds recorded at Luerwald, Amsberg, Germany | 71.18 | 11.7785 | 9.842 |
| human speech | Recording of Pilot announcement on flight to Amsterdam | 69.01 | 19.473 | 15.659 |

**Table 4.3:** *Properties of Audio Dataset. c is the regularity constant in the bounds, which here was calculated from each dataset split. $c'$ is the corresponding constant using the Bernstein-type bounds.*

For each of the images, we select the top-left corner of a 50x50 pixel window in each image uniformly at random and reshape to a vector with 2500 dimensions, repeating this one thousand times for each of the images. We then project the vectors using RS, orthonormalized Gaussian random projection (RP), Achlioptas sparse random projections (SRP) (with $P_{i,j} = \pm 1$ with probability $\frac{1}{6}$, and 0 with probability $\frac{2}{3}$), and also the first $k$ eigenvectors from applying PCA to the full sample of the one thousand vectors. The projected vectors were all scaled according to the values in Table 4.4. Note that a scaling correction for PCA was not employed in Bingham and Mannila (2001) where it was claimed that a straightforward rule is difficult to give. However, one can verify the *average* scaling for PCA projected vectors (over the dataset) in the squared Euclidean norm should be $\frac{Trace(\Sigma)}{Trace(\Sigma(1:k))}$ – that is the ratio of the trace of the covariance matrix of the data to the fraction of the trace retained under the PCA projection. We use the square root of this to approximately recover the correct scaling.

For the image data, this procedure was repeated for all twenty-three image files for each projection approach with a projection dimension $k$ range from 5 to 600 in increments of 5.

| Method | Norm Scaling Factor |
|---|---|
| Gaussian Random Projection | $\sqrt{\frac{d}{k}}$ |
| Sparse Random Projection | $\sqrt{\frac{1}{k}}$ |
| Random Subspace | $\sqrt{\frac{d}{k}}$ |
| Principal Component Analysis | $\sqrt{\frac{Trace(\Sigma)}{Trace(\Sigma(1:k))}}$ |

**Table 4.4:** *Theoretical norm-scaling quantities for the various projection schemes.*

For the DOROTHEA dataset for each of the three dataset splits, we first removed features with zero variance but, to avoid possible confounds, we carried out no other filtering. We then projected the data using RP, SRP, and RS as before with the projection dimension $k \in \{5, \ldots, 70,000\}$ for RS,

$k \in \{5, \ldots, 2,750\}$ for RP and SRP. For these data, the computational cost of PCA is prohibitive, and so we did not evaluate the effect of PCA projection on the DOROTHEA dataset. For the audio data, we took the left channel audio data, and then randomly took 1000 snippets of 44100 samples (1 second) with a random start time. We then projected the data using RP, SRP and RS as before with the projection dimensions $k \in \{5, \ldots, 2000\}$ for RS, RP, and SRP. We did not evaluate the effect of PCA projection on the audio dataset due to the computational cost of PCA.

For the three types of data, we randomly selected one hundred observations, and for each possible pair of these, we calculated the $\ell_2$ norm of the difference between the (scaled) projected observations $\|P(u-v)\|$ and the original points $\|u-v\|$. We then calculated the ratio between the (scaled) projected norm and the true norm $\frac{\|P(u-v)\|}{\|u-v\|}$ for each observation where the scaling constants used were those in Table 4.4.

For the image data, we plot in Figure 4.3 for each choice of $k$, the average of this ratio over all images as well as the 5-th and 95-th percentiles for the ratios. We also plot the runtime for the image data on the left in Figure 4.9 for each projection method versus $k$.

For DOROTHEA we repeated our experiments five times on each dataset split, to obtain an average over fifteen runs. As in the images, we report the mean ratio of the norms $\frac{\|P(u-v)\|}{\|u-v\|}$ as well as the 5th and 95th percentiles of this ratio in Figure 4.6. The average runtime for each different approach on Dorothea can be seen on the right in Figure 4.9. For the audio data, we plot in Figure 4.3 for each choice of $k$, the average of this ratio as well as the 5-th and 95-th percentiles for the ratios for each of the audio type. We also plot the runtime for the image data on the left in Figure 4.9 for each projection method versus $k$.

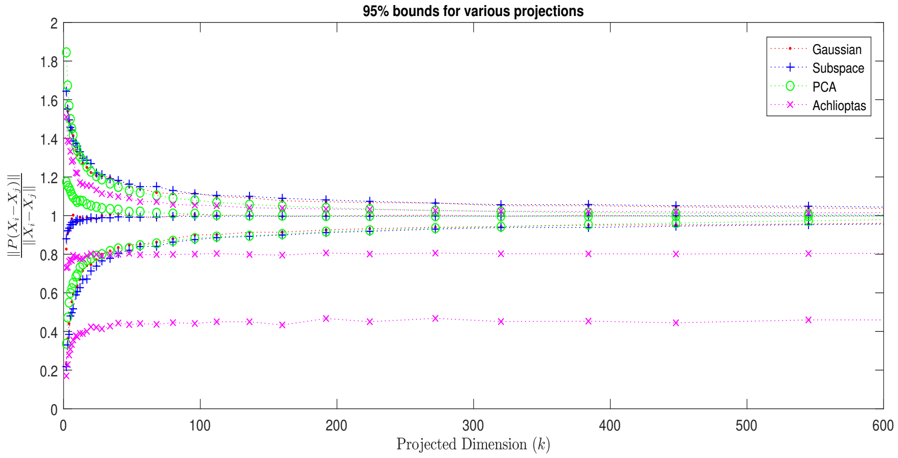

**Figure 4.3:** *Mean and 5th and 95th percentiles of the ratio* $\frac{\|P(X_i - X_j)\|}{\|X_i - X_j\|}$ *for image data vs. projection dimension* $k$*. We see that for* $k \gtrsim 80$ *Gaussian RP and RS are indistinguishable on these data. Note also the 5th percentile for SRP cf. Figure 4.4: Sparse RP frequently seems to underestimate norms on these data.*

## 4.4 Experimental Results and Discussion

Our experimental results corroborate our theory. We observe for natural image data that RS indeed gives a similar performance in terms of norm preservation to RP and, surprisingly, better performance than SRP on these data (as does RP) – (see Figures 4.4 and 4.3). Given the small values of $c$ estimated for these data (See Table 4.1) the similar performance to RP is broadly in line with what we would predict from theory; indeed Figure 4.3 shows that RS is nearly indistinguishable from the computationally more expensive RP on these data. On the other hand, one remarkable finding is that the distribution of norms for SRP is left-skewed here, and there is ample evidence that SRP consistently tends to underestimate distances between points when the correct theoretical scaling is applied, at least on these data. In this respect, SRP does worst on images such as the high contrast one above the centre column of Table 4.4, where we might instead reasonably expect RS to suffer from such a problem. Indeed, the normal distribution fit for RS applied to this image does show heavier tails for RS than for RP, but unlike SRP the error distribution is symmetric, and the centre of mass is in the right place at 1. We

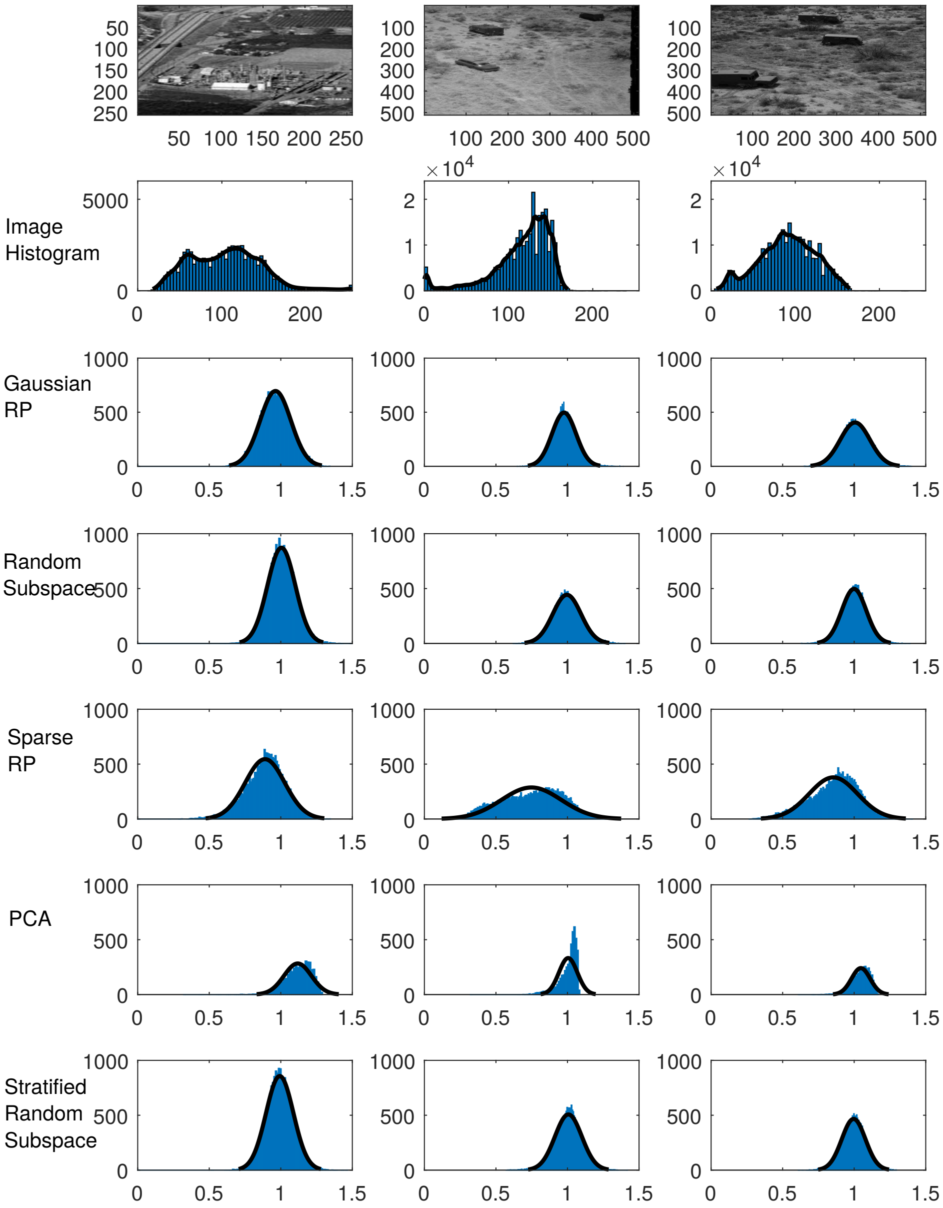

**Figure 4.4:** *Histograms of the ratio* $\frac{\|P(X_i - X_j)\|}{\|X_i - X_j\|}$ *for a fixed* $k = 50$ *dimensions on three representative images from the image dataset (i.e. data with small values of c or c') with overlaid normal density plots,* $n = 4950$*. Observe that RS has similar norms preservation performance as Gaussian RP on image data.*

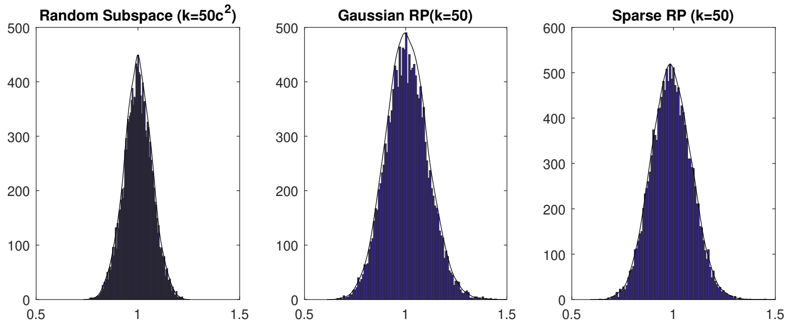

**Figure 4.5:** *Histograms of the ratio $\frac{\|P(X_i - X_j)\|}{\|X_i - X_j\|}$ for DOROTHEA dataset (i.e. data with large values of c or c') with RP with projection dimension $k_{rp} := k = 50$ for RP and SRP (middle and right plots) and comparison with RS with projection dimension $k_{rs} = 1750 > c' \times k_{rp}$ dimensions with overlaid normal density plots, $n = 4,950$. We see that errors behave nearly identically for RP and RS as predicted by theory.*

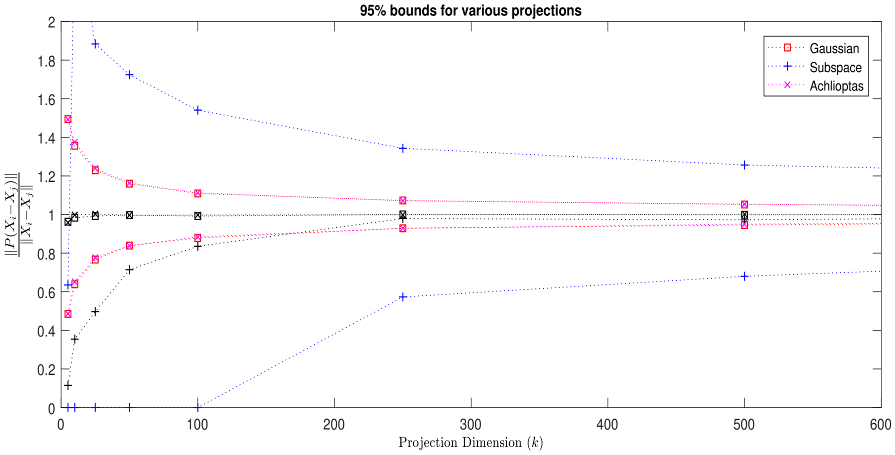

**Figure 4.6:** *Mean and 5th and 95th percentiles of the ratio $\frac{\|P(X_i - X_j)\|}{\|X_i - X_j\|}$ for Dorothea vs. projection dimension $k$. We see that for RS a much higher $k$ is required than for RP, though RS eventually catches up.*

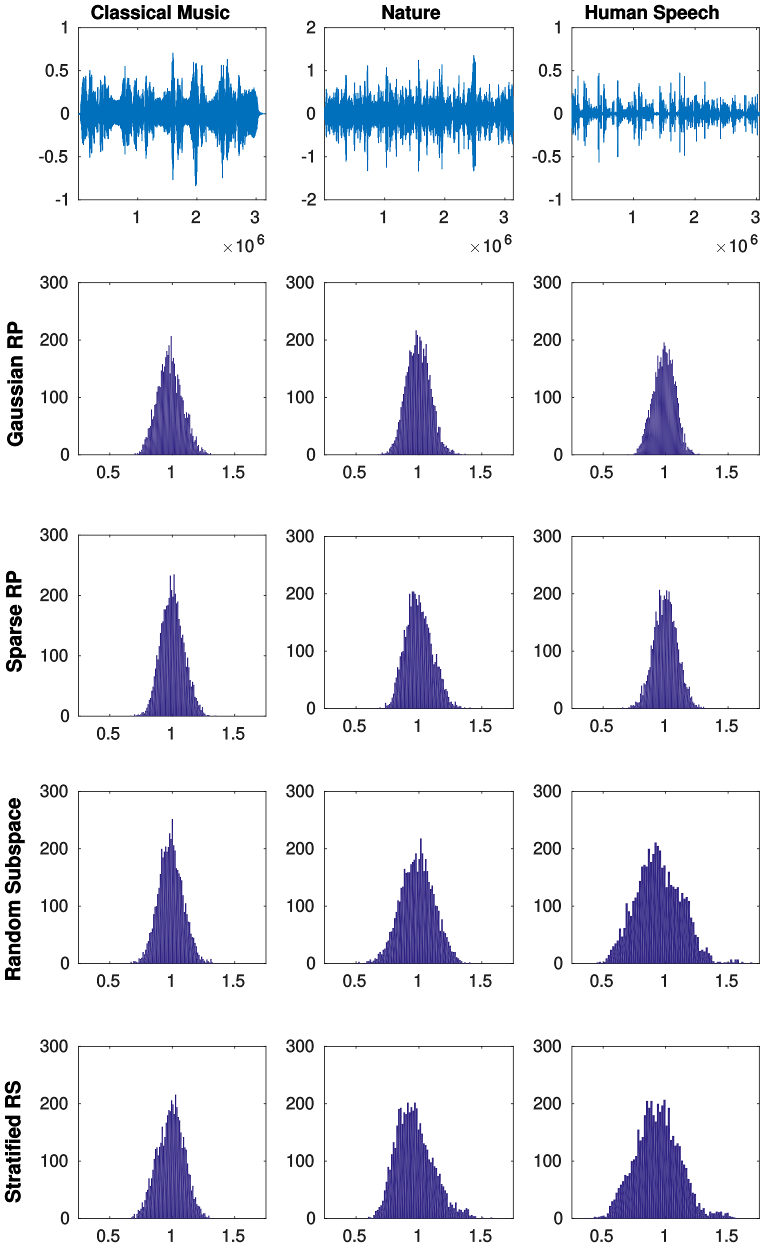

**Figure 4.7:** *Histograms of the ratio* $\frac{\|P(X_i - X_j)\|}{\|X_i - X_j\|}$ *for a fixed projection dimension* $k = 50$ *on the three audio files. density plots,* $n = 4950$*. Observe that RS has similar norm preservation performance as RP on classical music, and worse norm preservation performance than RP on human speech, as predicted by our theory.*

do not have a reasonable explanation for why SRP should be worse than RS on these images, but as we see clearly in Figure 4.3, this problem persists even as $k$ grows. A further interesting finding is that, unlike the results reported in Bingham and Mannila (2001), the performance of PCA scaled according to the scheme outlined in Table 4.4 is – for a large enough choice of $k$ – superior

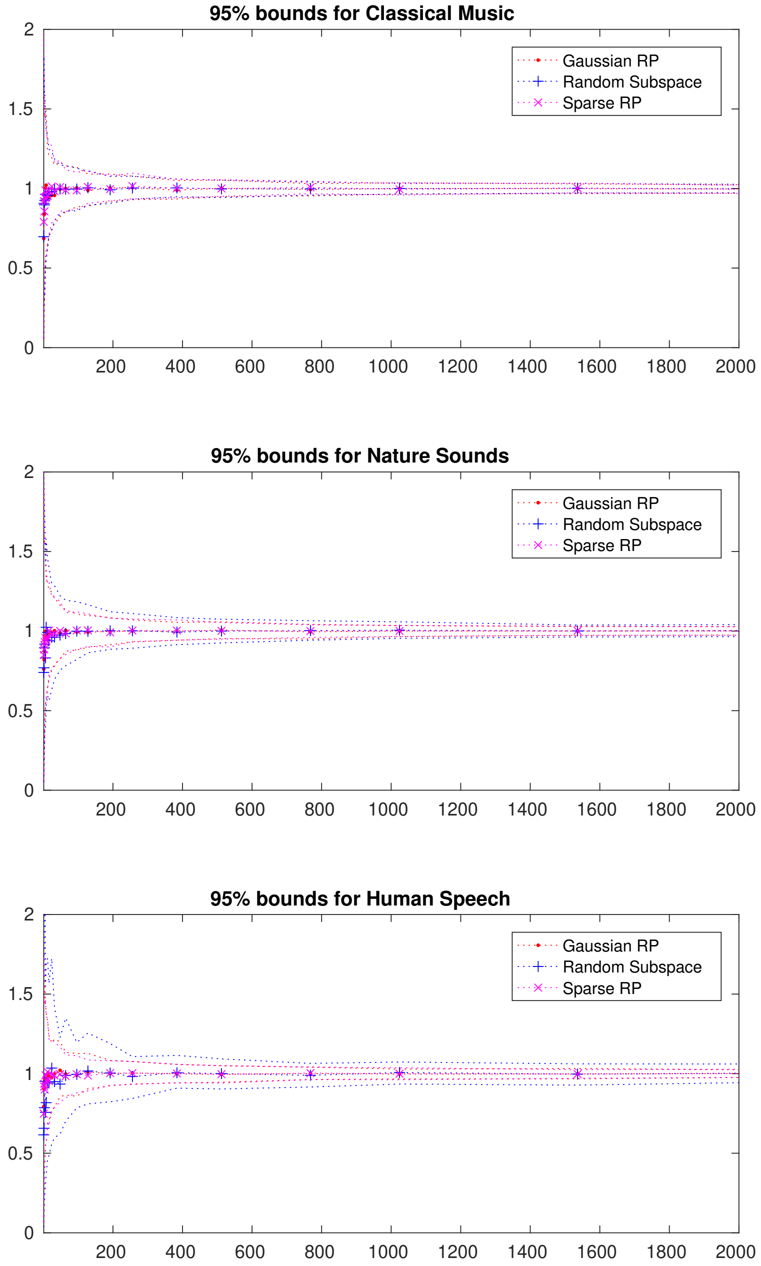

**Figure 4.8:** *Mean and 5th and 95th percentiles of the ratio* $\frac{\|P(X_i - X_j)\|}{\|X_i - X_j\|}$ *for audio data vs. projection dimension* $k$*. See that the performance of RS vs RP is almost indistinguishable for the classical music data, and RP performs significantly better than RS for human speech.*

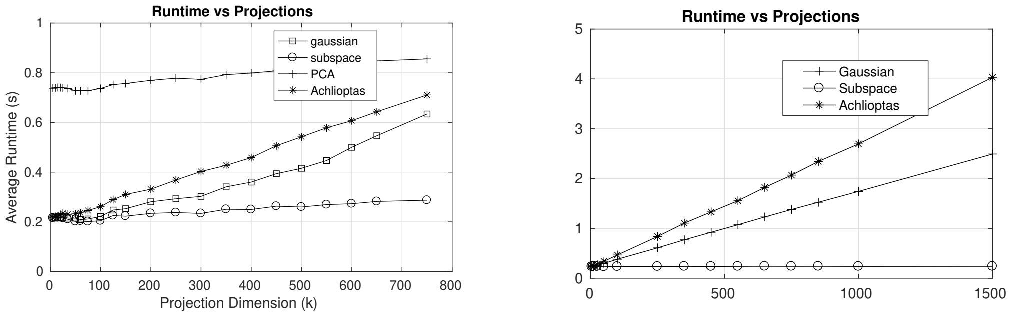

**Figure 4.9:** *(Left) Comparison of the runtime on dense image datasets with dimensionality $d = 2500$ vs projection dimension $k$. (Right) Comparison of the runtime on DOROTHEA with $d \simeq 100,000$ and proportion of non-zeros $\simeq 0.1$. Gaussian RP was faster than SRP here because generating the SRP matrix in MATLAB was slower for such large values of d than generating the Gaussian matrix.*

to all three random alternatives we have considered. The superior performance of PCA is to be expected since PCA maximises the retained within-feature variance on the projected sample and the scaling we proposed is adaptive in a non-linear way to this quantity, unlike the other alternatives which do not consider local properties of the data cloud and use a scaling that is linear in $k$. How far similar outcomes would hold for other types of data remains for future research. However, we note that it must depend on both the choice of $k$ and also on the rate at which the spectrum of the sample covariance matrix – the eigendecomposition of which gives the principal components – decays, since the scaling correction we apply to PCA is piecewise constant in $k$ with a non-uniform step size. We also note that, unlike for RP, SRP and RS, for PCA there is no theory to guide the user's choice of $k$ *a priori* even if one has access to the constant $c$ we require in our RS bounds.

Finally, we look at the computational cost of the different approaches considered: These are compared in Figure 4.9. For a fixed $k$ there is, of course, a significant runtime improvement in using RS compared to RP and SRP. On these data, it seems that choosing $k$ as the same for RS, RP, and SRP works equally well and so, everything else being equal, one would likely prefer RS to RP or SRP here. Note that in general, however, for fixed error, if $c \gg 8$ then the projection dimension $k$ for RS will be around $c^2$ times greater than for RP or SRP, so there is a trade-off. Whether one would prefer to use RS with a larger $k$ than for RP (for the same high-probability error guarantee) will depend on problem specifics such as the time complexity of the algorithm receiving the projected data with respect to the dimension, or whether it is more important to classify or to train quickly. Finally, PCA is, of course, computationally much more expensive when compared to the other three approaches, but we see that with the proper scaling term on these data it outperforms them in terms of geometry preservation. Thus, for PCA, there is essentially the same accuracy-vs-complexity trade-off as for RS.

The DOROTHEA data is very high-dimensional with only around 10% of entries non-zero, and for these data, the theory predicts that we will have poor norm preservation from RS compared to RP except when $k$ is very large. Our experimental results – see Figures 4.6 and 4.5 – show that indeed is the case. While RS does catch up with RP and SRP in terms of error eventually, both RP and SRP attain smaller error much more quickly than RS. On the other hand, we see in Figure 4.5 that after scaling the projected dimension required for RP by $c^2$ that RS indeed has comparable (and sometimes better) error performance than RP or SRP. We also see in Figure 4.6, that interestingly, unlike for the image data the scaled SRP does not tend to underestimate norms consistently, and all approaches (eventually) have their centre of mass at 1. Finally, despite the increased projection dimension, for a fixed error guarantee either variant of RS still gives us significantly improved runtime compared to RP and SRP (See Figure 4.9).

In the case of the audio data, we have three audio files with different regularity constants. The results for the audio data show that the norm preservation for RS lies somewhere in between the results for the image data and the DOROTHEA datasets. This result is in line with the predictions of our theory. Indeed, we also see that the spread in the norms is proportional with the estimated regularity constants, with tighter norm preservation on the classical music data, and looser norm preservation on the human speech data. – see Figures 4.8 and 4.7. As in the case with DOROTHEA, RS does eventually catch up with RP and SRP in terms of error with the classical music catching up at a faster rate and the human speech the slowest to catch up.

## 4.5 Feature Stratification

Theorem 4.3 suggests that an upper bound on $\Pr\left\{\left|\frac{d}{k}\left\|PX\right\|_2^2 - \left\|X\right\|_2^2\right| > \epsilon\right\}$ is similar to a Bernstein-Bennett-type bound. Interpreting the norm ratios in the argument of the $\exp(\cdot)$ in this bound as a variance in the squared vector components suggests that any sampling scheme that reduces this variance would imply a smaller failure probability $\delta$: For example by stratifying the features based on the similarity of their entries, we may obtain a tighter bound (or smaller $k$). This intuition comes from observing that $\frac{\|X\|_4^4}{\|X\|_2^4}$ is analogous to a second central moment in $X^2$. In order to explore this idea, we used $k-$means clustering with $m$ clusters to group the features of $X$ into strata with similar average values. We then sampled from these strata using 'Neyman allocation', i.e. using simple random sampling of the individual strata, we sampled from each stratum a number of features proportional to the contribution to the total variation in squared norm from that stratum, for a total of $k$ features. Our experimental results show that this approach typically does improve norm preservation under RS projection, given an appropriate choice of the number of strata. Although choosing the optimal number of strata $m$ is not completely straightforward since too small a value would not reduce variance appreciably, while too large a value could introduce additional noise and new sources of variation in the summand, we found that $m \approx \sqrt{k}$ seemed to work quite well generally. On the other hand, we note that the number of strata $m$ (and whether to expect improvement from stratifying at all) must be data-dependent, and by looking at the distribution over values across the features, e.g. by viewing the image histogram, we can still estimate the value that $m$ should take, at least approximately. For example, we observed on the natural image dataset that multi-modal or high contrast images benefitted more from having several strata; while, as one would expect unimodal or low contrast images did not benefit as much from a stratification approach. See Figures 4.4 and 4.10 for a visual comparison between the different approaches, while the table 4.5 shows the improvements in the 95% − 5% range and the standard deviation of $\|PX\|/\|X\|$ in the samples from stratified RS projection over the vanilla RS projection. Note that for image 7.1.02, stratification slightly increased the variability in $\|PX\|/\|X\|$.

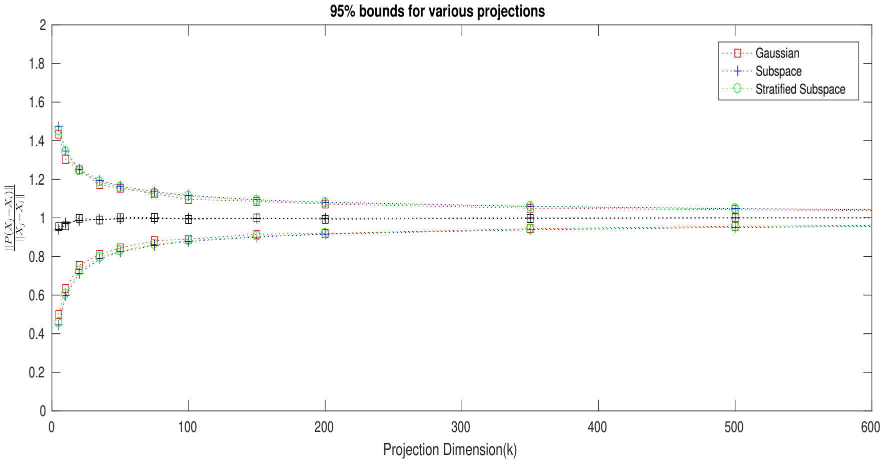

**Figure 4.10:** *Mean and 5th and 95th percentiles of the ratio* $\frac{\|P(X_i - X_j)\|}{\|X_i - X_j\|}$ *for image data vs. projection dimension* $k$ *for different number of clusters. We see that a stratified sampling reduces the spread of the interval compared to figure 4.3.*

## 4.6 Application to Compressive Sensing

RS has been a widely used approach in the literature for large-scale classification and regression problems, with many empirical studies confirming its effectiveness in these domains. Motivated by our theoretical findings, we decided to examine a possible new application of RS for compressive sensing since a JLL-type property in the sensing matrix[2] is key to sparse signal reconstruction using compressive measurements. In particular, we ask whether

[2] The 'restricted isometry property' or 'RIP'.

| File Name | 95%-5% range | | Standard Deviation | | Improvement over RS | |
|---|---|---|---|---|---|---|
| | m=1 | m=7 | m=1 | m=7 | range | SD |
| 5.1.09 | 0.3644 | 0.3062 | 0.1133 | 0.0933 | 15.97% | 17.65% |
| 5.1.10 | 0.2919 | 0.2481 | 0.0885 | 0.0766 | 15.01% | 13.45% |
| 5.1.11 | 0.6258 | 0.5922 | 0.189 | 0.1737 | 5.37% | 8.10% |
| 5.1.12 | 0.4146 | 0.2913 | 0.1237 | 0.1162 | 29.74% | 6.06% |
| 5.1.14 | 0.2957 | 0.2842 | 0.0894 | 0.0863 | 3.89% | 3.47% |
| 5.2.08 | 0.2854 | 0.2617 | 0.0907 | 0.0802 | 8.30% | 11.58% |
| 5.2.09* | 0.3975 | 0.3666 | 0.1233 | 0.1106 | 7.77% | 10.30% |
| 5.2.10 | 0.2975 | 0.2351 | 0.0911 | 0.0721 | 20.97% | 20.86% |
| 5.3.01* | 0.2465 | 0.2386 | 0.076 | 0.0715 | 3.20% | 5.92% |
| 5.3.02* | 0.4059 | 0.3328 | 0.1217 | 0.1106 | 18.01% | 9.12% |
| 7.1.01* | 0.3116 | 0.2916 | 0.1 | 0.089 | 6.42% | 11.00% |
| 7.1.02* | 0.5725 | 0.519 | 0.1671 | 0.1673 | 9.34% | -0.12% |
| 7.1.03 | 0.2879 | 0.2314 | 0.0864 | 0.0725 | 19.62% | 16.09% |
| 7.1.04 | 0.2929 | 0.2655 | 0.0893 | 0.0809 | 9.35% | 9.41% |
| 7.1.05* | 0.2366 | 0.2228 | 0.0717 | 0.0675 | 5.83% | 5.86% |
| 7.1.06 | 0.2948 | 0.2177 | 0.0898 | 0.0661 | 26.15% | 26.39% |
| 7.1.07* | 0.3002 | 0.2875 | 0.0904 | 0.0884 | 4.23% | 2.21% |
| 7.1.08* | 0.4339 | 0.3624 | 0.1368 | 0.1186 | 16.48% | 13.30% |
| 7.1.09 | 0.2753 | 0.2255 | 0.0839 | 0.0699 | 18.09% | 16.69% |
| 7.1.10 | 0.3326 | 0.2745 | 0.1006 | 0.0829 | 17.47% | 17.59% |
| 7.2.01* | 0.341 | 0.3151 | 0.1033 | 0.0987 | 7.60% | 4.45% |
| boat | 0.2996 | 0.2556 | 0.0893 | 0.077 | 14.69% | 13.77% |
| elaine | 0.2608 | 0.1995 | 0.0796 | 0.0593 | 23.50% | 25.50% |
| Average Improvement | | | | | 13.35% | 11.68% |

**Table 4.5:** *Comparison between stratified RS and vanilla RS. The table shows 95%-5% range and the standard deviation of the ratio* $\|PX\|/\|X\|$ *with* $k$ *fixed at 50, and either* $m = 7$ *strata or if the filename is marked with an asterisk (*)* $m = 6$ *strata.*

we can replace the dense matrix-matrix multiplication implied by a Gaussian sensing matrix for compressive sensing, with a far cheaper RS projection.

Results from Candes et al. (2004); Candes (2008), show that a sparse vector $x \in \mathbb{R}^D$ can be recovered from a small number of linear measurements by solving a convex programme. We used $\ell_1$-magic (Candes and Romberg, 2005), a MATLAB toolkit for compressive sensing for sparse reconstruction.

### 4.6.1 Theory

It is well known that images are highly compressible and can be represented by a "relatively" small number of coefficients without perceptible degradation in image quality. We intend to show that RS projection works as well as RP as a compressive sensing matrix for signal reconstruction if the regularity constant $c$ as defined in section 4.2 is small (i.e. $c < 4$).

According to Candes (2008) recovering $\boldsymbol{x}$ by minimizing $\min \|x\|_1$ subject to $\boldsymbol{Ax} = \boldsymbol{b}$ given that $\boldsymbol{x}$ is $s-$sparse

**Definition 4.1** (Restricted Isometry Property (Candes, 2008))**.** *For each integer $s = 1, 2, \ldots,$ define the isometry constant $\delta_s$ of a matrix $\boldsymbol{A}$ as the smallest number such that*

$$(1 - \delta_s)\|\boldsymbol{x}\|_2^2 \leq \|\boldsymbol{Ax}\|_2^2 \leq (1 + \delta_s)\|\boldsymbol{x}\|_2^2$$

*holds for all $s$-sparse vectors $\boldsymbol{x}$. A vector is said to be $s$-sparse if it has at most $s$ non-zero entries.*

**Theorem 4.7** (Noiseless Recovery (Candes, 2008))**.** *Assume that $\delta_{2s} < \sqrt{2} - 1$, Then the solution to $\boldsymbol{x}^*$ to $\boldsymbol{Ax} = \boldsymbol{b}$ obeys*

$$\|\boldsymbol{x}^* - \boldsymbol{x}\|_1 \leq C_0\|\boldsymbol{x} - \boldsymbol{x}_s\|_1$$

*and*

$$\|\boldsymbol{x}^* - \boldsymbol{x}\|_1 \leq s^{-1/2}C_0\|\boldsymbol{x} - \boldsymbol{x}_s\|_1$$

*for some constant $C_0$. In particular if $\boldsymbol{x}$ is $s$-sparse, the recovery is exact.*

The implications of Theorem 4.7 above shows that recovery of sparse signals using RS projection matrices as a compressive sensing matrix is possible

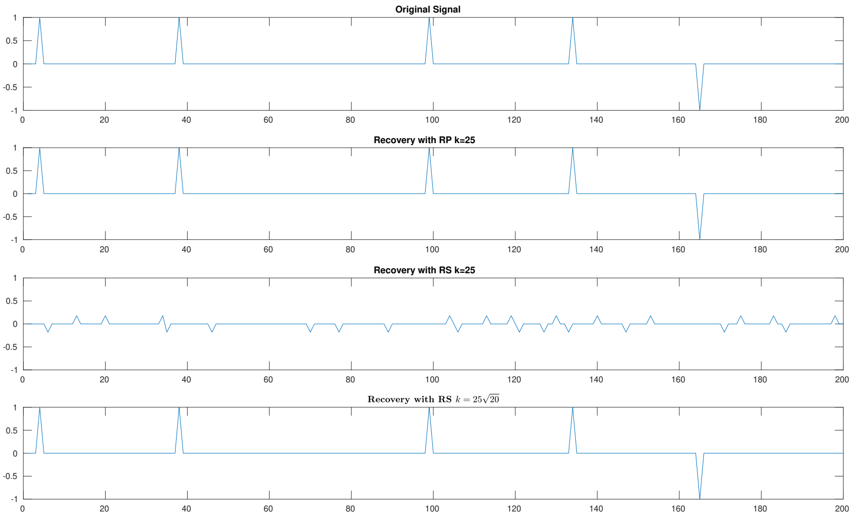

**Figure 4.11:** *Sparse reconstruction of* $\pm 1$ *signals using RS and RP. Note that RS requires a larger number of samples* ($k = 112 > 25c'$*, with* $c' = \sqrt{20}$) *to perfectly reconstruct the signal as implied by Theorem 4.4.*

provided the errors in the norms are not large. However, Theorem 4.4 show that the number of subspaces required for an error grows proportionally with the regularity constant which increases with the sparsity of the data. Having these two competing requirements seem to imply that RS is unsuitable as a RIP projection. In Figure 4.11, we see that for a given sparse signal recovery problem, using RS as a sensing matrix requires significantly more compressive samples to recover the signal. However, as what we have noted in our discussion in section 4.2.1 it is the "denseness" of the representation of the data that gives Johnson-Lindenstrauss-like (and in this context Restricted Isometry Property) norm preserving guarantees in RS projections.

Using this theory as our foundation and inspired by the results of Candes et al. (2004) on signal reconstruction using Fourier transforms, we constructed an experimental setup that shows the applicability of using RS for compressive sensing.

A Discrete Cosine Transform (DCT) transformation on a vector $\boldsymbol{x}$ with length $d$ can be represented by the product of a $d \times d$ orthogonal matrix $\boldsymbol{D}$ to the vector $\boldsymbol{x}$. A typical natural image can be represented by relatively few

DCT coefficients with most of the significant entries in the first few indices of the resulting vector. However, if we randomly shuffle $\boldsymbol{x}$ before applying the DCT transformation, the energy in coefficients of the DCT transformation would be spread out to the higher order coefficients giving a more "regular" representation that gives RP norm preserving guarantees.

Let $\boldsymbol{x}$ be the sparse vector containing the DCT coefficients of $\boldsymbol{X}$, let $\boldsymbol{D}$ be the matrix representing the DCT transform, let $\boldsymbol{S}$ be the permutation matrix representing shuffling of the pixels in $\boldsymbol{X}$ and $\boldsymbol{P}$ be the matrix representing the random subspace projection. Note that $\boldsymbol{P}, \boldsymbol{D}$ and $\boldsymbol{S}$ are orthogonal, and the inverses are simply the transpose of the respective matrices. The sequence of operations can therefore be written as $\boldsymbol{P}(\boldsymbol{D}\boldsymbol{S}\boldsymbol{D}^T)\boldsymbol{x}$ however since $\boldsymbol{D}^T\boldsymbol{x} = \boldsymbol{X}$ this can be simplified further to $\boldsymbol{P}(\boldsymbol{D}\boldsymbol{S}\boldsymbol{X})$. While it appears that we have costly matrix-matrix operations and there is no benefit to using RS over RP, the sequence of matrix multiplication with $\boldsymbol{P}$ and $\boldsymbol{S}$ can be implemented simply by addressing the different indices. Therefore, the matrix multiplications can be implemented in $O(1)$ if the image array fits in the memory and $O(d)$ if the array does not fit in memory. Moreover, the DCT transformations are often handled by specially optimised Digital Signal Processing routines and algorithms. Note that this is consistent with the theorems for sparse signal recovery in that the vector $\boldsymbol{x}$ that is recovered through the programme is sparse, however the representation $(\boldsymbol{D}\boldsymbol{S}\boldsymbol{D}^T\boldsymbol{x})$ we apply random subspace on is dense. A visual representation of this intuition is illustrated in Figure 4.12. [3]

### 4.6.2 Experimental Setup and Results

In our experiments, we compare RP and RS projected natural image data. Using the same images listed in table 4.1 and selecting the central 256x256-pixel crop of each image. We apply DCT on these central images. This gives

[3] In practice, we are applying a $2D$-DCT transform on a $d \times d$ image $\boldsymbol{X}$ before reshaping the $d \times d$ transformation to a $d^2$ vector. Analytically this is different from the series of matrix operations above; however, conceptually we can use the description above to describe the behaviour of our programme.

us a 65536-dimensional representation of the image with a very right-skewed coefficient distribution (i.e. in our context with a large value for $c$ or $c'$). We also apply an alternate DCT on the image after randomly shuffling the pixels, and this pixel-shuffled DCT representation gives us a representation that has components with a much 'denser' representation. (i.e. $c$ or $c'$ is reduced by this representation). Table 4.7 summarizes the values of $c'$ for each of the images. Note that the value for $c'$ in the dense representation is $\approxeq 4.9$ for all the images.

.

Figure 4.13 shows examples of the output from DCT with or without shuffling, with the larger DCT coefficients appearing lighter. These high-dimensional representations of the image were projected using Gaussian RP and RS with subspace dimensions $k = \{25, 100, 500, 2500\}$. We then use $\ell_1$-magic with quadratic ($\ell_2$ norm) constraints to recover the DCT coefficients and finally recover the images by inverting the pre-processing. Figure 4.12 shows a pictorial representation of the experimental setup.

We measure both the mean squared error between the recovered DCT coefficients and the original coefficients, and between the original and the recovered image. Figures 4.14 and 4.15 gives a visual summary of images reconstructed from RP and RS with varying levels of compression.

From our experiments, we found that RS projections give similar performance to RP projections in terms of residual squared error when applied on a dense DCT representation. As suggested by our theory, RS does not perform as well when it is applied to a sparse DCT representation. One significant advantage of RS over RP is that RS remains significantly faster than RP as shown in Figure 4.16 even with a higher projection dimension, $k$, whereas RP takes significantly longer. We also experienced memory issues with RP when applied with a large projection dimension.

Overall, based on these outcomes RS for compressive sensing seems to show some promise, at least for image data. We also tried the same experimental

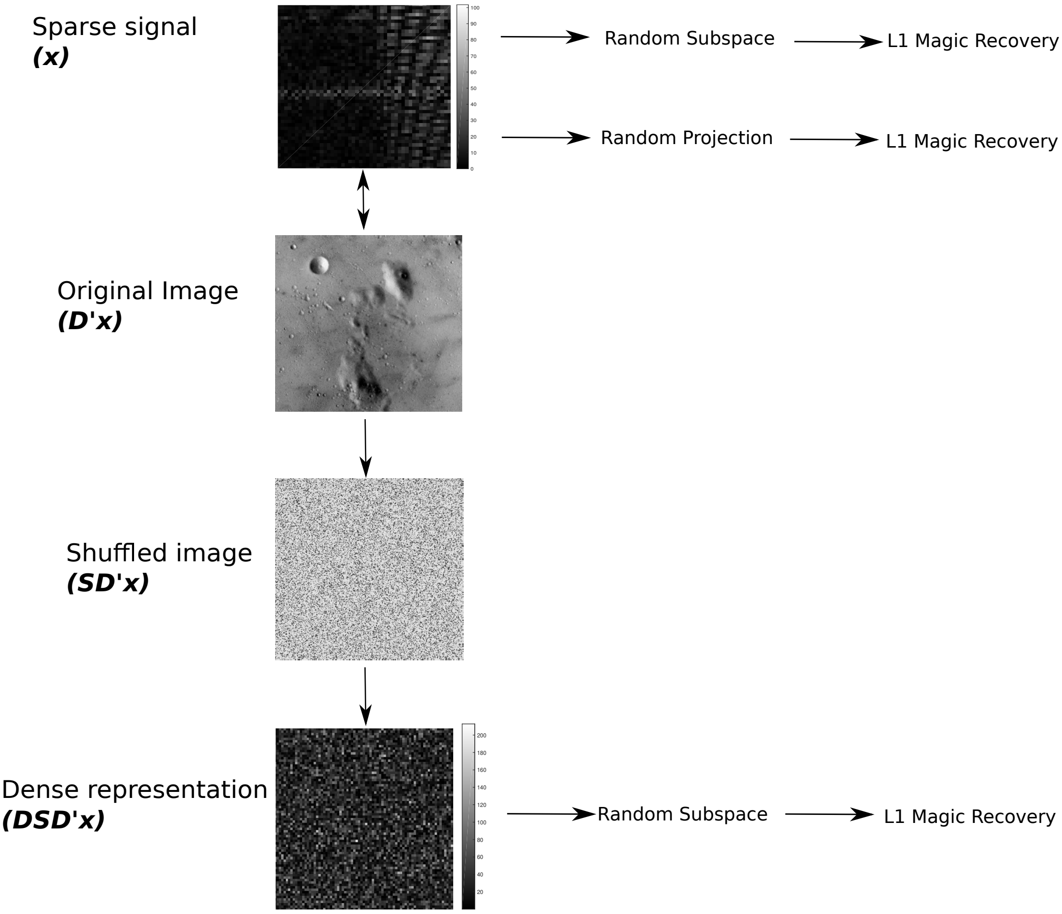

**Figure 4.12:** *Pictorial representation of the signal recovery experimental setup.* $\boldsymbol{x}$ *is the sparse signal to be recovered, and* $\boldsymbol{DSD'x}$ *is the dense representation, and* $RS$ *as the sampling matrix to successfully recover* $\boldsymbol{x}$*.*

setup with audio data with mixed results. While $\ell_1$-magic was able to recover the audio waveform with some degree of fidelity, we observed that there were noticeable white noise and audible distortion in the recovered audio. Our results appear to be consistent with the demonstration by Balzano et al. (2010).

Additional note: We also observed that using RS with our experimental setup works with the TV-EQ ($\ell_1$-norm equality) algorithm in the $\ell_1$-magic toolbox, whereas RP tends to run into convergence issues on the TV-EQ algorithm. It is straight-forward to modify the proof for Theorem 4.4 for $\ell_1$-norm preservation guarantees.

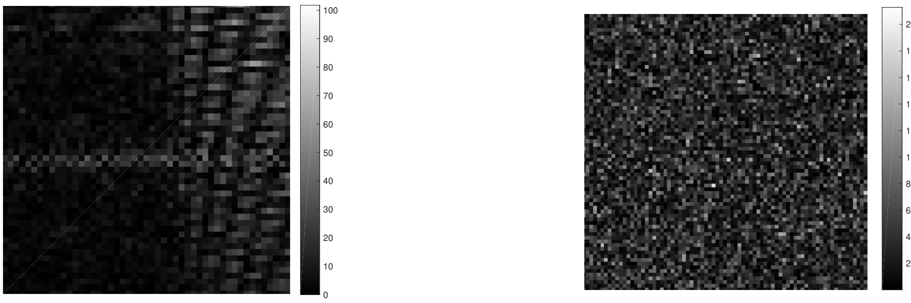

**Figure 4.13:** *Visual representation of the DCT of an unshuffled image (left) and shuffled image (right). Observe that DCT coefficients of the unshuffled image is "sparse", and the DCT coefficients of the shuffled image is "dense".*

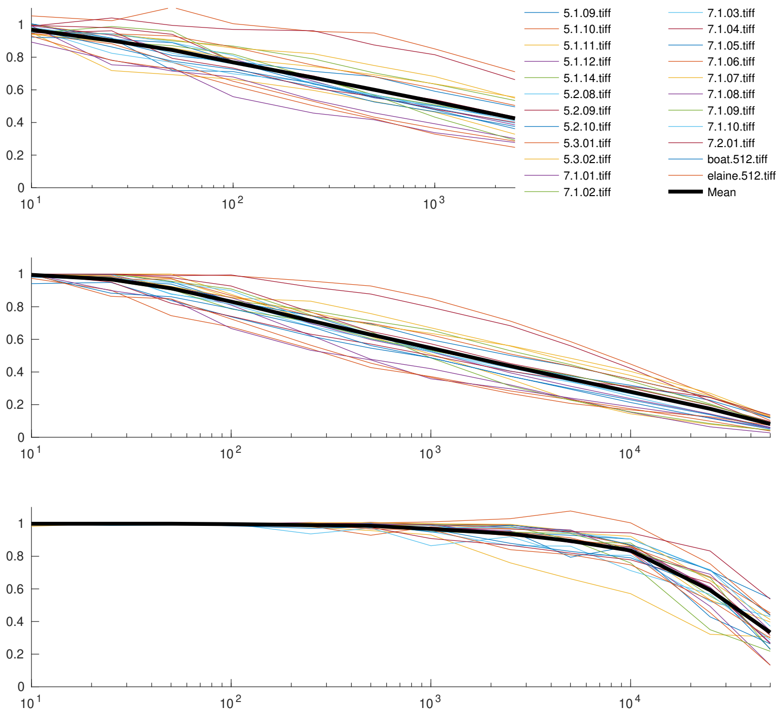

**Figure 4.14:** *Individual mean squared errors in reconstructed images, for compressive sensing of image data from Gaussian RP (Top) and Random Subspace projections plus shuffled DCT (Middle) and unshuffled DCT (Bottom) versus the number of compressive samples (projection dimension). Note that the horizontal axis is in log scale. Each coloured line is a separate image and the bold line is the average from table 4.6 over all the images.*

|  | % Mean Square Error |  |  |
|---|---|---|---|
| Subspaces | Gaussian RP | Dense RS | Sparse RS |
| 10 | 96.82% | 99.53% | 99.93% |
| 25 | 90.15% | 96.76% | 99.94% |
| 50 | 84.40% | 91.29% | 99.95% |
| 100 | 76.85% | 83.13% | 99.68% |
| 250 | 67.45% | 71.40% | 99.08% |
| 500 | 60.07% | 62.90% | 98.68% |
| 1000 | 52.69% | 54.70% | 96.77% |
| 2500 | 42.49% | 43.63% | 93.73% |
| 5000 |  | 35.67% | 89.40% |
| 10000 |  | 27.85% | 83.53% |
| 25000 |  | 17.45% | 59.22% |
| 50000 |  | 8.13% | 33.30% |

**Table 4.6:** *Average mean squared error over all the reconstructed images for Random Subspace projection versus the projection dimension $k$.*

|  | c' |
|---|---|
| 5.1.09.tiff | 170.23 |
| 5.1.10.tiff | 34.95 |
| 5.1.11.tiff | 114.58 |
| 5.1.12.tiff | 165.36 |
| 5.1.14.tiff | 98.32 |
| 5.2.08.tiff | 97.94 |
| 5.2.09.tiff | 40.43 |
| 5.2.10.tiff | 181.37 |
| 5.3.01.tiff | 215.66 |
| 5.3.02.tiff | 90.79 |
| boat.512.tiff | 118.79 |
| elaine.512.tiff | 144.08 |

|  | c' |
|---|---|
| 7.1.01.tiff | 107.46 |
| 7.1.02.tiff | 120.16 |
| 7.1.03.tiff | 134.65 |
| 7.1.04.tiff | 73.66 |
| 7.1.05.tiff | 130.83 |
| 7.1.06.tiff | 98.88 |
| 7.1.07.tiff | 133.90 |
| 7.1.08.tiff | 247.96 |
| 7.1.09.tiff | 131.18 |
| 7.1.10.tiff | 128.49 |
| 7.2.01.tiff | 189.68 |
| Average | 138.25 |

**Table 4.7:** *Estimated $c'$ for unshuffled DCT representation. Note that $c'$. For the dense DCT representation $c' \approxeq 4.9$ for all of the images.*

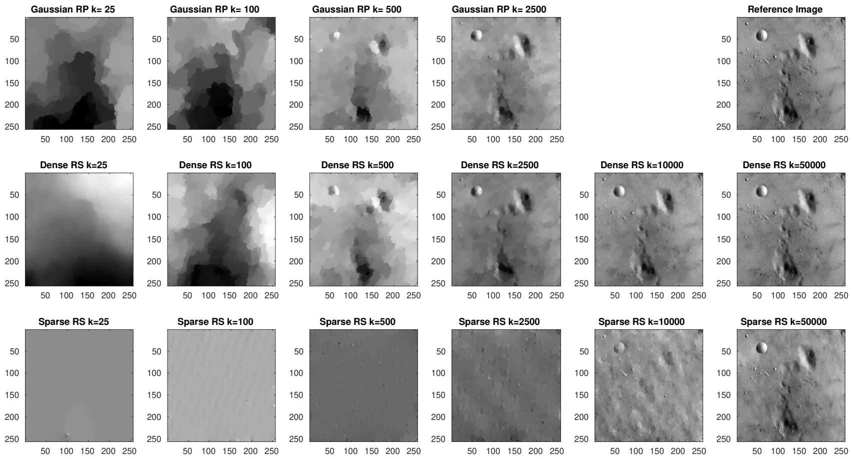

**Figure 4.15:** *Visual comparison between images reconstructed using compressive sensing with RP and RS as the sensing approach for image 5.1.09. The top row shows recovered images for DCT plus RP, the second row for DCT with pixel-shuffling plus RS, the bottom row for DCT without pixel-shuffling plus RS. The original image is in the top right-hand corner. Results for other images were similar.*

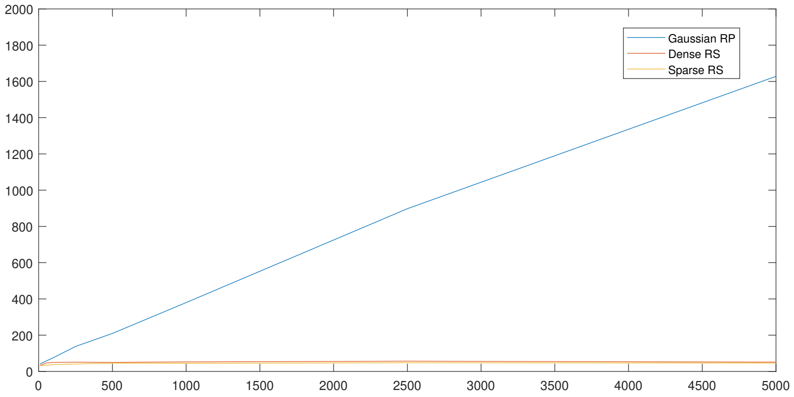

**Figure 4.16:** *End to end runtime of $\ell_1$-Magic for RP and RS applied to shuffled and unshuffled DCT representation vs number of compressive samples. Observe that the runtime for RP grows linearly with the number of compressive samples at a much faster rate than RS*

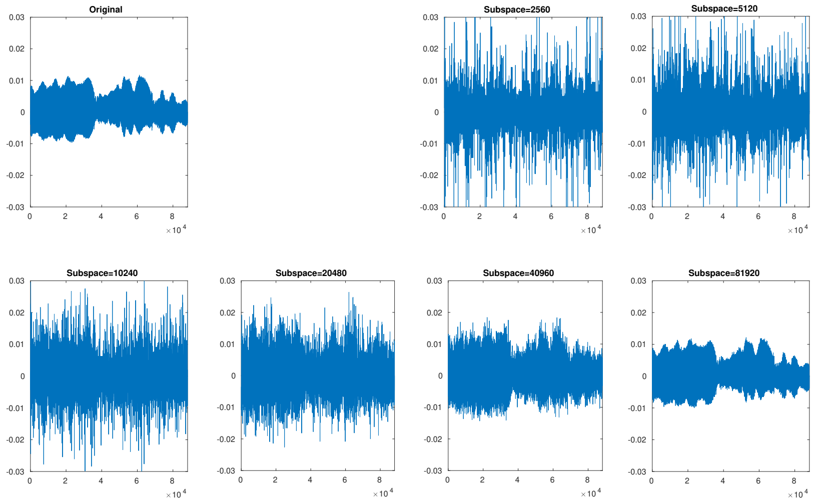

**Figure 4.17:** *Visual comparison between audio reconstructed using compressive sensing with RS as the sensing approach for audio file "Danse Arabe". Observe that there the reconstructed audio has significant amount of white noise at low number of compressive samples, which gradually improve with higher number of compressive samples. Subjectively, the audio snippet was recognizable at* 10240 *compressive samples.*

## 4.7 Conclusion and Summary

In this chapter, we have shown that the guarantees for norm-preservation in random subspace projection are dependent on the regularity of the features in the data. We have defined a regularity measure that can be used to determine the number of subspaces needed for a given error in the norm.

We corroborated our theories empirically and showed that for regular data such as natural images, random subspace could achieve geometry preservation performance comparable to random projection but with significant runtime improvement. We also demonstrated that sampling schemes that reduce variance such as stratification could improve on the norm preserving properties of random subspace. While some additional computation cost may be incurred from clustering the data, if the strata are not known *a priori*, we can reduce the average error in the squared Euclidean norms through stratification. Using the results of Arriaga and Vempala (1999), we showed that with high probability a robust half-space classifier concept can be learned as a direct consequence of the JLL-like guarantees.

Finally, we noted that for compressive sensing, a sensing matrix with the so-called Restricted Isometry Property furnished guarantees for perfect reconstruction of a signal from a compressed sample of it and showed how RS could be used practically for compressive sensing.

In the next chapter, in light of the dot-product preservation properties of RS, we will derive the flipping probability of RS projections and the theoretical performance guarantees of RS projected classifiers in the absence of a margin. Additionally, in chapter 6, we will investigate the performance of RS ensembles in the light of our theories developed in this chapter with emphasis on the regularity constant $c'$ on the number of projection dimensions needed to build a reliable RS ensemble.

# 5

# Flip Probabilities of Random Subspace Projected Vectors

**Summary** In chapter 4, we noted that the norm preservations guarantee provides a margin-dependent generalization bounds for RS projections (Arriaga and Vempala, 1999). Following from the work of Durrant and Kabán (2013), we will provide generalization bounds for RS classification in the absence of margin using "Flipping probabilities". Flipping probability is defined as the probability that two vectors in $d$-dimensions with an angular separation of less than $\pi/2$ have an angular separation more than $\pi/2$ after projecting to a lower dimensional space. Our approach is to derive the sub-Gaussian norms of RS projected vectors then use the upper bound from both to bound the generalization error. We will also demonstrate that for RS, unlike RP, the probabilistic guarantees for "flipping probability" are data-dependent and depends on the $\ell_4$ and $\ell_\infty$ norms of the data vectors and the classifier.

Leveraging this dependence on the data representation, we propose a computationally efficient transformation that can significantly improve the upper bound on the flipping probability for very sparse vectors in subsection 5.2.3.

In section 5.3, we corroborate our theoretical findings empirically on synthetic data and discuss the limitations of these probabilistic bounds on the flipping probability. Moreover, in section 5.4, we discuss the practical implications of our results for classification ensembles.

## 5.1 Background

Durrant and Kabán (2010) defined the "flipping probability" of a pair of randomly projected vectors as the probability that two vectors in $d-$dimensional Euclidean space $\boldsymbol{m}, \boldsymbol{n} \in \mathbb{R}^d$ which are separated in $\mathbb{R}^d$ by an angle $\theta_{\boldsymbol{m},\boldsymbol{n}} \in [0, \pi/2]$ to have angular separation $\theta_{\boldsymbol{R},\boldsymbol{m},\boldsymbol{n}} > \pi/2$ following a random projection. Later work by Durrant and Kabán (2013) shows that the flipping probability is a useful tool to capture the geometric structure that makes a classification problem "easy" in the sense that it requires a relatively small sample size to guarantee good generalisation.

Durrant and Kabán (2013) showed that the generalization error w.r.t the (0,1)-loss of any linear classifier can be bounded by a function of the flipping probability in the following theorem

**Theorem 5.1** (ERM Generalization Error of RP projected data sets (Theorem 3.1 Durrant and Kabán (2013))). *Let $\mathcal{T}^n = \{(\boldsymbol{x}^{(i)}, y^{(i)}) | \boldsymbol{x}^{(i)} \in \mathbb{R}^d, y^{(i)} \in \{0,1\}\}_{i=1}^n$ be a set of $d$-dimensional labelled training examples of size $n$ and let $\hat{h}$ be the linear ERM classifier estimated from $\mathcal{T}^n$. Let $\boldsymbol{R} \in \mathcal{M}_{k \times d}$, $k < d$ be a random projection matrix with entries $R_{i,j} \overset{i.i.d}{\sim} N(0, \sigma)$. Denote by $\mathcal{T}^n_{\boldsymbol{R}} = \{(\boldsymbol{R}\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^n$ the RP projection of the training data $\mathcal{T}^n$, and let $\hat{h}_{\boldsymbol{R}}$ be the linear classifier estimated from $\mathcal{T}^n_{\boldsymbol{R}}$. Let $f_k(\theta_i)$ be the flipping probability following a random projection. Then, for all $\delta \in (0,1]$, with probability at least $1 - 2\delta$, w.r.t the random choice of $\mathcal{T}^n$ and $\boldsymbol{R}$, the generalization error of $\hat{h}_{\boldsymbol{R}}$ w.r.t. the $(0,1)$-loss is bounded above by:*

$$
\begin{aligned}
Pr\left\{\hat{h}_{\boldsymbol{R}}(\boldsymbol{R}\boldsymbol{x}^{(q)}) \neq y^{(q)}\right\} &\leq \hat{E}(\mathcal{T}^n, \hat{h}) + \frac{1}{n}\sum_{i=1}^{n} f_k(\theta_i) \\
&+ \min\left\{\sqrt{3\log\frac{1}{\delta}}\sqrt{\frac{1}{n}\sum_{i=1}^{n} f_k(\theta_i)}, \frac{1-\delta}{\delta}\frac{1}{n}\sum_{i=1}^{n} f_k(\theta_i)\right\} \\
&+ 2\sqrt{\frac{k+1\log\frac{2\epsilon n}{k+1} + \log\frac{1}{\delta}}{n}}
\end{aligned}
\tag{5.1}
$$

Where on the right hand side, the first term is the empirical risk of the classifier $\hat{h} \in \mathbb{R}^d$, the second is the empirical flipping probability measured on the data and the last two terms bound the deviation of the empirical estimates from their expectation with high probability.

For a random projection matrix $\boldsymbol{R}$ with zero mean, sub-Gaussian entries, Kabán and Durrant (2017) showed that the flipping probability can be bounded above by the following lemma

**Lemma 5.2** (Flipping probability upper bound, sub-Gaussian case (Lemma 1.3 Kabán and Durrant (2017))**.** *Let $\boldsymbol{R}$ be a RP matrix with entries $R_{i,j}$ drawn i.i.d from a zero mean sub-Gaussian distribution, let $\boldsymbol{h}, \boldsymbol{x} \in \mathbb{R}^d$ and let $\theta = \theta_{\boldsymbol{x}}^{\boldsymbol{h}}$ be the angle between them. Let $\boldsymbol{Rh}, \boldsymbol{Rx} \in \mathbb{R}^k$ be the images of $\boldsymbol{h}, \boldsymbol{x}$ under $\boldsymbol{R}$. Then, if $\boldsymbol{h}^T\boldsymbol{x} \neq 0$, we have:*

$$Pr\left\{\frac{(\boldsymbol{Rh})^T\boldsymbol{Rx}}{\boldsymbol{h}^T\boldsymbol{x}} \leq 0\right\} \leq \exp\left(-k\cos^2\theta/8\right)$$

Moreover, when $\boldsymbol{R}$ is a Gaussian random projection (i.e. $\boldsymbol{R}_{i,j} \overset{i.i.d}{\sim} N(0,\sigma^2)$), Durrant and Kabán (2010) used the fact that the Gaussian random projection is rotational invariant to show that the density function of the flipping probability for Gaussian RP projected vectors has the following form

$$\Pr\left\{\theta_{R,\boldsymbol{n},\boldsymbol{m}} > \pi/2 | \theta_{\boldsymbol{n},\boldsymbol{m}} < \pi/2\right\} = \frac{\int_0^{\theta_{\boldsymbol{n},\boldsymbol{m}}} \sin^{k-1}(\phi)d\phi}{\int_0^{\pi} \sin^{k-1}(\phi)d\phi}$$

where $\theta_{R,\boldsymbol{n},\boldsymbol{m}}$ is the angle between vectors $\boldsymbol{m}$ and $\boldsymbol{n}$ after a random projection and $\theta_{\boldsymbol{n},\boldsymbol{m}}$ is the original angle between vectors $\boldsymbol{m}$ and $\boldsymbol{n}$.

While the "flipping probability" is exactly known for RP, the flipping probability of Random Subspace (RS) projected vectors is not known. The proofs for the flipping probabilities of RP vectors exploit the rotation-invariant nature of the projection. However, RS projections are not rotationally invariant, and no guarantees for the flipping probability for RS currently exist.

## 5.2 Flipping Probability Bounds for RS projected vectors

In this section, we present our main theoretical results. First, we will upper-bound the sub-Gaussian norms of RS projected vectors to show that the empirical distribution of a random subspace projected vector is sub-Gaussian. Then we use the upper bound of Kabán and Durrant (2017) for the flipping probability of a sub-Gaussian distribution.

Now, recall from Section 3.3, general Hoeffding's inequality (Lemma 3.6),

**Lemma 5.3** (General Hoeffding's inequality, (Theorem 2.6.3 Vershynin (2018)))**.** *Let $X_1, \ldots, X_n$ be independent mean zero, sub-Gaussian random variables, and $\boldsymbol{a} = (a_1, \ldots a_n)^T \in \mathbb{R}^n$. Then for every $t \geq 0$, we have*

$$Pr\left\{\left|\sum_{i=1}^{N} a_i X_i\right| > t\right\} \leq 2\exp(-\frac{ct^2}{K^2\|\boldsymbol{a}\|_2^2}) \textit{ where } K = \max_i \|X_i\|_{\psi_2}$$

We will now show that the empirical distribution of RS projected vectors is sub-Gaussian, and therefore the flipping probability has a similar form as Lemma 5.2.

**Lemma 5.4.** *Let $\boldsymbol{u}$ and $\boldsymbol{v}$ be two unit vectors in $\mathbb{R}^d$ and let $\boldsymbol{P}$ be a random subspace projection chosen without replacement from $\mathbb{R}^d \mapsto \mathbb{R}^k$, with $k, d \in \mathbb{N}$ and $0 < k < d/2$, then the empirical distribution of $(\boldsymbol{Pu})^T\boldsymbol{Pv}$ is also sub-Gaussian.*

*Proof.* We want to show that

$$\Pr\left\{\left|\frac{d}{k}(\boldsymbol{Pu})^T\boldsymbol{Pv}) - \boldsymbol{u}^T\boldsymbol{v}\right| > t\right\} \leq 2\exp\left(\frac{-t^2}{K_1^2}\right)$$

for some constant $K_1$ and thus showing that the empirical distribution of $\frac{d}{k}(\boldsymbol{Pu})^T\boldsymbol{Pv} - \boldsymbol{u}^T\boldsymbol{v}$ is sub-Gaussian by Definition 3.15(1).

First observe that

$$\begin{aligned}\Pr\left\{\left|\frac{d}{k}(\boldsymbol{Pu})^T\boldsymbol{Pv} - \boldsymbol{u}^T\boldsymbol{v}\right| > t\right\} &= \Pr\left\{\left|\frac{d}{k}(\boldsymbol{Pu})^T\boldsymbol{Pv} - \frac{d}{k}\mathrm{E}[(\boldsymbol{Pu})^T\boldsymbol{Pv}]\right| > t\right\} \\ &= \Pr\left\{\left|(\boldsymbol{Pu})^T\boldsymbol{Pv} - \mathrm{E}[(\boldsymbol{Pu})^T\boldsymbol{Pv}]\right| > \frac{k}{d}t\right\}\end{aligned} \tag{5.2}$$

Using union bound, we can upper bound equation 5.2 by

$$\Pr\left\{(\boldsymbol{Pu})^T\boldsymbol{Pv} - \mathrm{E}[(\boldsymbol{Pu})^T\boldsymbol{Pv}] > \frac{k}{d}t\right\} + \Pr\left\{\mathrm{E}[(\boldsymbol{Pu})^T\boldsymbol{Pv}] - (\boldsymbol{Pu})^T\boldsymbol{Pv} < -\frac{k}{d}t\right\}$$

As we did in the proof for Theorem 4.1 in section A.1, we will let $I$ be the index set such that $i \in I \implies \boldsymbol{P}_{ii} = 1$. Observe that,

$$\Pr\left\{(\boldsymbol{Pu})^T\boldsymbol{Pv} - \mathrm{E}[(\boldsymbol{Pu})^T\boldsymbol{Pv}] > \frac{k}{d}t\right\} = \Pr\left\{\sum_{i\in I} u_iv_i > \frac{k}{d}(t + \sum_{i=1}^{d} u_iv_i)\right\}$$

Where the sample total $\sum_{i\in I} u_iv_i$ is estimated from a sample size of $k$ without replacement. Let $\boldsymbol{x} = \boldsymbol{u} \odot \boldsymbol{v}$, where $\odot$ is the Hadamard product operator. Observe that $x_i = u_iv_i, \forall i \in [1, d]$. Also observe that $\|\boldsymbol{x}\|_\infty = \|\boldsymbol{u} \odot \boldsymbol{v}\|_\infty \leq \|\boldsymbol{u}\|_\infty\|\boldsymbol{v}\|_\infty \leq 1$. We apply Hoeffding's inequality, giving us the following results:

$$\begin{aligned}\Pr\left\{\sum_{i\in I} u_iv_i > \frac{k}{d}(t + \sum_{i=1}^{d} u_iv_i)\right\} &= \Pr\left\{\sum_{i\in I} \boldsymbol{x}_i > \frac{k}{d}(t + \sum_{i=1}^{d} \boldsymbol{x}_i)\right\}\\ &\leq \exp\left(\frac{-2(\frac{kt}{d})^2}{\sum_{i\in I}\|\boldsymbol{x}\|_\infty^2}\right) \qquad (5.3)\\ &= \exp\left(\frac{-2k(\frac{t}{d})^2}{\|\boldsymbol{x}\|_\infty^2}\right)\end{aligned}$$

Similarly, we can find an upper bound to

$$\Pr\left\{\mathrm{E}[(\boldsymbol{Pu})^T\boldsymbol{Pv}] - (\boldsymbol{Pu})^T\boldsymbol{Pv} < -\frac{k}{d}t\right\} \leq \exp\left(\frac{-2k(\frac{t}{d})^2}{\|\boldsymbol{x}\|_\infty^2}\right) \qquad (5.4)$$

Applying union bound, an upper bound to equation 5.2 is

$$\Pr\left\{\left|\frac{d}{k}(\boldsymbol{Pu})^T\boldsymbol{Pv} - \boldsymbol{u}^T\boldsymbol{v}\right| > t\right\} \leq 2\exp\left(\frac{-2k(\frac{t}{d})^2}{\|\boldsymbol{x}\|_\infty^2}\right)$$

Therefore by Definition 3.15(1), The empirical distribution $\frac{d}{k}(\boldsymbol{Pu})^T\boldsymbol{Pv} - \boldsymbol{u}^T\boldsymbol{v}$ is sub-Gaussian with $K_1^2 = \frac{d^2 \max_i \|\boldsymbol{x}^{(i)}\|_\infty^2}{2k} \leq \frac{d^2}{2k}$ □

**Lemma 5.5** (Flipping probability upper bound, Random Subspace). *Let $\boldsymbol{P}$ be a RS projection chosen without replacement from $\mathbb{R}^d \mapsto \mathbb{R}^k$, with $k, d \in \mathbb{N}$ and $0 < k < d/2$, let $\boldsymbol{h}, \boldsymbol{x} \in \mathbb{R}^d$ be two vectors with unit length, and let $\theta = \theta_{\boldsymbol{x}}^{\boldsymbol{h}}$ be the angle between them. Let $\boldsymbol{Ph}, \boldsymbol{Px} \in \mathbb{R}^k$ be the images of $\boldsymbol{h}, \boldsymbol{x}$ under $\boldsymbol{P}$. Then, if $\boldsymbol{h}^T\boldsymbol{x} \neq 0$, we have:*

$$Pr\left\{\frac{(\boldsymbol{Ph})^T\boldsymbol{Px}}{\boldsymbol{h}^T\boldsymbol{x}} \leq 0\right\} \leq 2\exp\left(-2k\cos^2\theta/d^2\right)$$

*Proof.* Observe that

$$\begin{aligned}\Pr\left\{\frac{(\boldsymbol{Ph})^T\boldsymbol{Px}}{\boldsymbol{h}^T\boldsymbol{x}} \le 0\right\} &= \Pr\left\{\frac{(\boldsymbol{Ph})^T\boldsymbol{Px}}{\boldsymbol{h}^T\boldsymbol{x}} - \frac{k}{d} \le -\frac{k}{d}\right\} \\ &= \Pr\left\{\frac{d}{k}(\boldsymbol{Ph})^T\boldsymbol{Px} - \boldsymbol{h}^T\boldsymbol{x} \le -\boldsymbol{h}^T\boldsymbol{x}\right\} \qquad (5.5) \\ &\le \Pr\left\{\left|\frac{d}{k}(\boldsymbol{Ph})^T\boldsymbol{Px} - \boldsymbol{h}^T\boldsymbol{x}\right| \ge \boldsymbol{h}^T\boldsymbol{x}\right\}\end{aligned}$$

Substituting $t = \boldsymbol{h}^T\boldsymbol{x} = \cos\theta$ into equation 5.3 gives

$$\Pr\left\{\frac{(\boldsymbol{Ph})^T\boldsymbol{Px}}{\boldsymbol{h}^T\boldsymbol{x}} \le 0\right\} \le 2\exp\left(-\frac{2k\cos^2\theta}{d^2}\right)$$

□

A direct implication of our result is that by Theorem 5.1 the generalization error of a classifier trained by ERM on a RS projected data set can be upper bounded by

$$\begin{aligned}&\Pr\left\{\hat{h}_R(R\boldsymbol{x}^{(q)} \neq y^{(q)})\right\} \le \hat{\mathrm{E}}(\mathcal{T}^n, \hat{h}) + 2\exp\left(-\frac{k\cos^2\theta}{d^2}\right) \\ &+ \min\left\{\sqrt{3\log\frac{1}{\delta}}\sqrt{2\exp\left(-\frac{2k\cos^2\theta}{d^2}\right)}, \frac{1-\delta}{\delta}2\exp\left(-\frac{2k\cos^2\theta}{d^2}\right)\right\} \qquad (5.6) \\ &+ 2\sqrt{\frac{k+1\log\frac{2\epsilon n}{k+1} + \log\frac{1}{\delta}}{n}}\end{aligned}$$

One key observation here is that the generalization error bounds grows with the dimensionality of the original data unlike in the case for RP. We also would like to note however, that we used an overestimate of the sub-Gaussian norms, and in practice, the required projection dimensions would be lower than the data agnostic guarantees.

### 5.2.1 Data Dependent Flipping Probability

We have not imposed any regularity conditions on the projected vectors in the error bounds and generalization bound above. However, as the results of Chapter 4 shows, the norm and dot-product preservation guarantees (and by extension, geometry-preservation guarantees) of a random subspace projected vector depends on the 'regularity' of the representation of the vectors. By imposing some regularity conditions on the vectors, we can tighten the flipping probability bounds given by lemma 5.4 significantly. Here, we will use Bernstein's inequality to upper-bound the flipping probability

**Theorem 5.6** (Flip probability of random subspace projected vectors)**.** *Let $\boldsymbol{u}$ and $\boldsymbol{v}$ be two unit-vectors in $\mathbb{R}^d$ with an angular separation of $0 \leq \theta_{\boldsymbol{u},\boldsymbol{v}} \leq \pi$ and $\theta_{\boldsymbol{u},\boldsymbol{v}} \neq \pi/2$. Let $\boldsymbol{P}$ be a random subspace projection chosen without replacement from $\mathbb{R}^d \mapsto \mathbb{R}^k$, with $k \in [1, d)$, The "flipping probability" is upper-bounded by*

$$Pr\left\{\frac{(\boldsymbol{P}\boldsymbol{u})^T\boldsymbol{P}\boldsymbol{v}}{\boldsymbol{u}^T\boldsymbol{v}} \leq 0\right\} \leq \exp\left(\frac{-ck\cos^2\theta}{2d\|\boldsymbol{u}\odot\boldsymbol{v}\|_2^2}\right)$$

*where $c$ is a constant*

*Proof.* We first note that the flipping probability can also be expressed as below

$$\begin{aligned}\Pr\left\{\frac{(\boldsymbol{P}\boldsymbol{u})^T\boldsymbol{P}\boldsymbol{v}}{\boldsymbol{u}^T\boldsymbol{v}} < 0\right\} &= \Pr\left\{-\frac{(\boldsymbol{P}\boldsymbol{u})^T\boldsymbol{P}\boldsymbol{v}}{\boldsymbol{u}^T\boldsymbol{v}} > 0\right\} \\ &= \Pr\left\{-\frac{(\boldsymbol{P}\boldsymbol{u})^T\boldsymbol{P}\boldsymbol{v}}{\boldsymbol{u}^T\boldsymbol{v}} + \frac{k}{d} > \frac{k}{d}\right\}\end{aligned} \tag{5.7}$$

Now, observe that for the case $0 \leq \theta_{\boldsymbol{u},\boldsymbol{v}} < \pi/2$ we have $u^Tv > 0$ and

$$\begin{aligned}\Pr\left\{\frac{(\boldsymbol{P}\boldsymbol{u})^T\boldsymbol{P}\boldsymbol{v}}{\boldsymbol{u}^T\boldsymbol{v}} < 0\right\} &= \Pr\left\{\frac{k}{d}\boldsymbol{u}^T\boldsymbol{v} - (\boldsymbol{P}\boldsymbol{u})^T\boldsymbol{P}\boldsymbol{v} > \frac{k}{d}\boldsymbol{u}^T\boldsymbol{v}\right\} \\ &= \Pr\left\{\mathrm{E}[(\boldsymbol{P}\boldsymbol{u})^T\boldsymbol{P}\boldsymbol{v}] - (\boldsymbol{P}\boldsymbol{u})^T\boldsymbol{P}\boldsymbol{v} > \frac{k}{d}\cos\theta_{\boldsymbol{u},\boldsymbol{v}}\right\}\end{aligned} \tag{5.8}$$

Conversely when $\pi/2 < \theta_{\boldsymbol{u},\boldsymbol{v}} \leq \pi$, we have $u^Tv < 0$ and we have

$$\begin{aligned}\Pr\left\{\frac{(\boldsymbol{P}\boldsymbol{u})^T\boldsymbol{P}\boldsymbol{v}}{\boldsymbol{u}^T\boldsymbol{v}} < 0\right\} &= \Pr\left\{(\boldsymbol{P}\boldsymbol{u})^T\boldsymbol{P}\boldsymbol{v} - \frac{k}{d}\boldsymbol{u}^T\boldsymbol{v} > -\frac{k}{d}\boldsymbol{u}^T\boldsymbol{v}\right\} \\ &= \Pr\left\{(\boldsymbol{P}\boldsymbol{u})^T\boldsymbol{P}\boldsymbol{v} - \mathrm{E}[(\boldsymbol{P}\boldsymbol{u})^T\boldsymbol{P}\boldsymbol{v}] > -\frac{k}{d}\cos\theta_{\boldsymbol{u},\boldsymbol{v}}\right\}\end{aligned} \tag{5.9}$$

In the following steps, we will upper bound equations 5.8 and 5.9.

As in Lemma 5.4, we let $\boldsymbol{x} = \boldsymbol{u} \odot \boldsymbol{v}$. We now set $\boldsymbol{q}$ to be a proxy for $E[\boldsymbol{P}] - \boldsymbol{P}$. Let $q_i$ for all $i \in [1, d]$ be i.i.d Bernoulli random variables such that

$$\boldsymbol{q}_i := \begin{cases} -1 + \frac{k}{d} & \text{w.p. } \frac{k}{d} \\ \frac{k}{d} & \text{w.p. } \frac{d-k}{d} \end{cases}$$

Observe that $\boldsymbol{q}_i$ is an independent zero mean random variable. Observe also $\mathrm{Var}(\sum_{i=1}^{d} \boldsymbol{q}_i \boldsymbol{x}_i) = \mathrm{Var}(\boldsymbol{q}^T \boldsymbol{x}) = \frac{d-k}{d}\frac{k}{d}\|\boldsymbol{x}\|_2^2 \leq \frac{k}{d}\|\boldsymbol{x}\|_2^2$

By Lemma 3.9, we can transfer the probability for sampling with replacement into the setting of sampling without replacement. Observe that an upper bound for equation 5.8 is

$$\Pr\left\{\mathrm{E}[(\boldsymbol{Pu})^T\boldsymbol{Pv}] - (\boldsymbol{Pu})^T\boldsymbol{Pv} > \frac{k}{d}\cos\theta_{\boldsymbol{u},\boldsymbol{v}}\right\} \leq \Pr\left\{\sum_{i=1}^{d} q_i x_i > \frac{k}{d}\cos\theta_{\boldsymbol{u},\boldsymbol{v}}\right\} \tag{5.10}$$

Applying Bernstein's inequality we have,

$$\Pr\left\{\sum_{i=1}^{d} q_i x_i \geq t\right\} \leq \exp\frac{-\frac{1}{2}t^2}{\frac{k}{d}\|\boldsymbol{x}\|_2^2 + \frac{1}{3}\|\boldsymbol{x}\|_\infty t}$$

We choose $t = \frac{k}{d}\cos\theta_{\boldsymbol{u},\boldsymbol{v}}$ and observing that $\exp\left(\frac{-\frac{1}{2}t^2}{\frac{k}{d}\|\boldsymbol{x}\|_2^2+\frac{1}{3}\|\boldsymbol{x}\|_\infty t}\right) \leq \exp\left(\frac{-\frac{1}{2}\frac{k}{d}\cos^2\theta_{\boldsymbol{u},\boldsymbol{v}}}{2(\|\boldsymbol{x}\|_2^2)}\right)$, with the last inequality from observing that $\|\boldsymbol{x}\|_\infty \leq \|\boldsymbol{x}\|_2$.

Now for equation 5.9, we let $\boldsymbol{x} = \boldsymbol{u} \odot \boldsymbol{v}$. We now set $\boldsymbol{q}$ to be a proxy for $\boldsymbol{P} - E[\boldsymbol{P}]$. Let $q_i$ for all $i \in [1, d]$ be i.i.d Bernoulli random variables such that

$$\boldsymbol{q}_i := \begin{cases} 1 - \frac{k}{d} & \text{w.p. } \frac{k}{d} \\ -\frac{k}{d} & \text{w.p. } \frac{d-k}{d} \end{cases}$$

Observe that as in the previous case, $\boldsymbol{q}_i$ is an independent zero mean random variable. Observe also $\mathrm{Var}(\sum_{i=1}^{d} \boldsymbol{q}_i \boldsymbol{x}_i) = \mathrm{Var}(\boldsymbol{q}^T \boldsymbol{x}) = \frac{d-k}{d}\frac{k}{d}\|\boldsymbol{x}\|_2^2 \leq \frac{k}{d}\|\boldsymbol{x}\|_2^2$

Again, by Lemma 3.9, we can transfer the probability for sampling with replacement into the setting of sampling without replacement and observing that an upper bound for equation 5.9 is

$$\Pr\left\{\mathrm{E}[(\boldsymbol{Pu})^T\boldsymbol{Pv}] - (\boldsymbol{Pu})^T\boldsymbol{Pv} > \frac{k}{d}\cos\theta_{\boldsymbol{u},\boldsymbol{v}}\right\} \leq \Pr\left\{\sum_{i=1}^{d} q_i x_i > \frac{k}{d}\cos\theta_{\boldsymbol{u},\boldsymbol{v}}\right\} \tag{5.11}$$

Applying Bernstein's inequality we have,

$$\Pr\left\{\sum_{i=1}^{d} q_i x_i \geq t\right\} \leq \exp\frac{-\frac{1}{2}t^2}{\frac{k}{d}\|\boldsymbol{x}\|_2^2 + \frac{1}{3}\|\boldsymbol{x}\|_\infty t}$$

We choose $t = -\frac{k}{d}\cos\theta_{\boldsymbol{u},\boldsymbol{v}} > 0$ and observing that $\exp\left(\frac{-\frac{1}{2}t^2}{\frac{k}{d}\|\boldsymbol{x}\|_2^2+\frac{1}{3}\|\boldsymbol{x}\|_\infty t}\right) \leq \exp\left(\frac{-\frac{1}{2}\frac{k}{d}\cos^2\theta_{\boldsymbol{u},\boldsymbol{v}}}{2(\|\boldsymbol{x}\|_2^2)}\right)$.

Observe that the upper bounds are the same for both cases where $u^T v > 0$ and $u^T v < 0$. Therefore we have,

$$\Pr\left\{\frac{(\boldsymbol{P}\boldsymbol{u})^T\boldsymbol{P}\boldsymbol{v}}{\boldsymbol{u}^T\boldsymbol{v}} < 0\right\} \leq \exp\left(\frac{-\frac{1}{2}\frac{k}{d}\cos^2\theta_{\boldsymbol{u},\boldsymbol{v}}}{2(\|\boldsymbol{x}\|_2^2)}\right)$$

□

We now consider the properties of $\|\boldsymbol{u}\odot\boldsymbol{v}\|_2^2$. We let $\boldsymbol{u}^\perp$ be the orthogonal component of $\boldsymbol{v}$ such that $\boldsymbol{v} = \boldsymbol{u}\cos\theta_{\boldsymbol{u},\boldsymbol{v}} + \boldsymbol{u}^\perp\sin\theta_{\boldsymbol{u},\boldsymbol{v}}$. Now, observe that we can rewrite $\|\boldsymbol{u}\odot\boldsymbol{v}\|_2^2$ as

$$\begin{aligned}
\|\boldsymbol{u}\odot\boldsymbol{v}\|_2^2 &= \sum_{i=1}^d u_i^2(v_i)^2 \\
&= \sum_{i=1}^d u_i^2(u_i\cos\theta_{\boldsymbol{u},\boldsymbol{v}} + u_i^\perp\sin\theta_{\boldsymbol{u},\boldsymbol{v}})^2 \\
&= \sum_{i=1}^d u_i^4\cos^2\theta_{\boldsymbol{u},\boldsymbol{v}} + 2u_i^3u_i^\perp\sin\theta_{\boldsymbol{u},\boldsymbol{v}}\cos\theta_{\boldsymbol{u},\boldsymbol{v}} + u_i^2(u_i^\perp)^2\sin^2\theta_{\boldsymbol{u},\boldsymbol{v}} \\
&= \|\boldsymbol{u}\|_4^4\cos^2\theta_{\boldsymbol{u},\boldsymbol{v}} + \|\boldsymbol{u}\odot\boldsymbol{u}^\perp\|_2^2\sin^2\theta_{\boldsymbol{u},\boldsymbol{v}} + 2\|\boldsymbol{u}^3\odot\boldsymbol{u}^\perp\|\sin\theta_{\boldsymbol{u},\boldsymbol{v}}\cos\theta_{\boldsymbol{u},\boldsymbol{v}}
\end{aligned} \tag{5.1}$$

Now if we replace equation 5.1 into Theorem 5.6, the flipping probability of vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ under a RS projection can be written as

$$\Pr\left\{\frac{(\boldsymbol{P}\boldsymbol{u})^T\boldsymbol{P}\boldsymbol{v}}{\boldsymbol{u}^T\boldsymbol{v}} \leq 0\right\} \leq \exp\left(\frac{-ck}{2d\|\boldsymbol{u}\|_4^4 + \|\boldsymbol{u}\odot\boldsymbol{u}^\perp\|_2^2\tan^2\theta_{\boldsymbol{u},\boldsymbol{v}} + 2\|\boldsymbol{u}^3\odot\boldsymbol{u}^\perp\|\tan\theta_{\boldsymbol{u},\boldsymbol{v}}}\right)$$

This implies that the distribution of flipping probability has a minimum, independent of the angle of the vectors that is determined by the $\ell_4$ norm of one of the vector. We will like to note that under most circumstance, $\|\boldsymbol{u}^3\odot\boldsymbol{u}^\perp\|$ is typically very small unless $\boldsymbol{u}$ is heavily skewed, e.g. few large positive entries with many small negative entries or vice versa. In cases where $\|\boldsymbol{u}^3\odot\boldsymbol{u}^\perp\|$ is significant, the distribution of the flipping probability according to Theorem 5.6 will also be skewed.

**Corollary 5.7.** *Let $\boldsymbol{u}$ and $\boldsymbol{v}$ be two unit vectors in $\mathbb{R}^d$ with an angular separation of $\theta_{\boldsymbol{u},\boldsymbol{v}}$. Let $\boldsymbol{P}$ be a random subspace projection chosen without replacement from $\mathbb{R}^d \mapsto \mathbb{R}^k$, with $0 < k < d/10$, Let $\boldsymbol{u}^\perp$ be the orthogonal component of $\boldsymbol{v}$ such that $\boldsymbol{v} = \boldsymbol{u}\cos\theta_{\boldsymbol{u},\boldsymbol{v}} + \boldsymbol{u}^\perp\sin\theta_{\boldsymbol{u},\boldsymbol{v}}$. The upper bound of the flipping probability is given by*

$$Pr\left\{\boldsymbol{u}^T\boldsymbol{P}\boldsymbol{v} < 0|\boldsymbol{u}^T\boldsymbol{v} > 0\right\} < \exp\left(\frac{-ck}{d(\|\boldsymbol{u}\|_4^4 + \|\boldsymbol{u}\|_\infty^2\tan^2\theta_{\boldsymbol{u},\boldsymbol{v}} + 2\|\boldsymbol{u}\|_6^3\tan\theta_{\boldsymbol{u},\boldsymbol{v}})}\right)$$

*where $c$ is a constant*

*Proof.* Applying Holder's inequality to $\|\boldsymbol{u} \odot \boldsymbol{u}^{\perp}\|_2^2$, we have

$$\|\boldsymbol{u} \odot \boldsymbol{u}^{\perp}\|_2^2 = \sum_{i=1}^{d} u_i^2 (u_i^{\perp})^2 \leq \left(\sum_{i=1}^{d} u_i^{2p}\right)^{1/p} \left(\sum_{i=1}^{d} (u_i^{\perp})^{2q}\right)^{1/q} \quad \text{with } 1/p + 1/q = 1$$

Choosing $q = 1$ and $p = \infty$, and observing that $\sum_{i=1}^{d} (u_i^{\perp})^2 = \|\boldsymbol{u}^{\perp}\|_2^2 = 1$ gives us $\|\boldsymbol{u} \odot \boldsymbol{u}^{\perp}\|_2^2 \leq \|\boldsymbol{u}\|_{\infty}^2$.

Finally, applying Holder's inequality to $\|\boldsymbol{u}^3 \odot \boldsymbol{u}^{\perp}\|_1$ we have,

$$\|\boldsymbol{u}^3 \odot \boldsymbol{u}^{\perp}\|_1 = \sum_{i=1}^{d} u_i^3 (u_i^{\perp}) \leq \left(\sum_{i=1}^{d} |u_i^{3p}|\right)^{1/p} \left(\sum_{i=1}^{d} |(u_i^{\perp})^q|\right)^{1/q} \quad \text{with } 1/p + 1/q = 1$$

. Choosing $p = 2$ and $q = 2$ we have $\|\boldsymbol{u}^3 \odot \boldsymbol{u}^{\perp}\| \leq \|\boldsymbol{u}\|_6^3$ and completing the proof. □

### 5.2.2 Discussion of the Bounds

Theorem 5.6 and Corollary 5.7 implies that the flipping probability of the angle between two vectors can be upper-bounded by the $\ell_4$ and $\ell_6$ and the $\ell_{\infty}^2$ norms of any one of vectors. When this is applied to linear classifiers, the flipping probability is the upper bound that an observation is misclassified after a RS projection is applied. Our theorems suggest that this probability can be upper-bounded by the norms of the normal vector of the discriminating hyperplane.

To gain some intuition of the data dependency, if we consider a Fisher's Linear Discriminant classifier for a binary classification problem with balanced class i.e. $(n_0 = n_1)$ centered on the origin (i.e. $\mu_0 + \mu_1 = 0$). We let $\mu_0$ and $\mu_1$ be the centres of observations belonging to class 0 and 1 respectively and $\Sigma = \sum_{j=0}^{1} (X_j - \mu_j)(X_j - \mu_j)^T$ be the covariance matrix of the training data and let $\lambda_i$ be the $i$-th eigenvalue of $\Sigma$ corresponding to the $i$-th coordinate. The normal vector of the discriminating hyperplane would then be $\boldsymbol{h} = \Sigma^{-1/2}(\mu_0 - \mu_1)$.

Letting $\boldsymbol{u} = \boldsymbol{h}/\|\boldsymbol{h}\|_2$, we see that $d\|\boldsymbol{u}\|_4^4 = \frac{d \sum_{i=1}^{d} \frac{(\mu_0 - \mu_1)_i^4}{\lambda_i^2}}{(\sum_{i=1}^{d} \frac{(\mu_0 - \mu_1)_i^2}{\lambda_i})^2}$, $d\|\boldsymbol{u}\|_{\infty}^2 = \frac{d \max_i \frac{(\mu_0 - \mu_1)_i^2}{\lambda_i}}{(\sum_{i=1}^{d} \frac{(\mu_0 - \mu_1)_i^2}{\lambda_i})^2}$. We see that $\|\boldsymbol{u}\|_4^4$ decreases as the vector $\boldsymbol{u}$ becomes more "regular" (i.e. the squared entries of $\boldsymbol{u}$ are close to each other). Observe that this is analogous to the regularity constant $c'$ derived in the previous chapter. Also, the flipping probability for "hard" classification problems (that is to say $\theta_{\boldsymbol{u},\boldsymbol{h}} \sim \pi/2$) increases the classification largely depends on only a few features. i.e. the weights for the discriminating hyperplane have few entries that are much larger than the average entries. Observe that this condition is analogous to the regularity constant $c$ defined in chapter 4.

### 5.2.3 Vector Densification using Householder transforms

Corollary 5.7 implies that for a pair of unit vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d$, we only need to minimize the fourth norm ($\ell_4$), and the squared infinity-th norm ($\ell_\infty^2$) of either one of the vector in order to improve the upper bound of the flipping probability. It is not difficult to see that the unit vector $\boldsymbol{u}' = \left(\pm\boldsymbol{1}/\sqrt{d}, \ldots, \pm\boldsymbol{1}/\sqrt{d}\right)$ indeed has the smallest $\ell_4$ and $\ell_\infty^2$ norm (the values are $\|\boldsymbol{u}\|_4^4 = \frac{1}{d}$ and $\|\boldsymbol{u}\|_\infty^2 = \frac{1}{d}$. Hence, if we can find a transformation that "regularize" $\boldsymbol{u}$ (i.e. to $\boldsymbol{u}'$) we can significantly improve on the upper bound flipping probability for vectors that has large $\|\boldsymbol{u}\|_4^4$ and $\|\boldsymbol{u} \odot \boldsymbol{u}^\perp\|_2^2$.

One such transformation that achieves this is the Householder transformation. A Householder transform $\boldsymbol{H}$ is given by $\boldsymbol{H} := \boldsymbol{I} - 2\boldsymbol{n}\boldsymbol{n}^T$ where $\boldsymbol{I}$ is the identity matrix and $\|\boldsymbol{n}\|_2 = 1$. One can easily check that $\boldsymbol{n}$ is an eigenvector of $\boldsymbol{H}$ with one of the eigenvalues $-1$, and all other eigenvalues are 1, and that $\boldsymbol{H} = \boldsymbol{H}^T = \boldsymbol{H}^{-1}$. Geometrically, $\boldsymbol{H}$ is therefore a reflection about a hyperplane through the origin with normal vector $\boldsymbol{n}$ and, in particular, $\ell_2$ norms are preserved by $\boldsymbol{H}$: $\|\boldsymbol{H}\boldsymbol{u}\|_2 = \|\boldsymbol{u}\|_2$ for any $\boldsymbol{u}$. Moreover $\boldsymbol{H}\boldsymbol{u} = \boldsymbol{u} - 2\boldsymbol{n}(\boldsymbol{n}^T\boldsymbol{u})$, so one need not evaluate the matrix multiplication explicitly.

We can determine the normal vector $\boldsymbol{n}$ to do a reflection of $\boldsymbol{u}$ to $\boldsymbol{u}'$ (or to arbitrary unit vector for that matter) in $O(d)$. One such algorithm to do this is given in appendix C and provides us with a very efficient method in which to 'densify' the vectors.

**Lemma 5.8** (Angular Preservation of Householder Transform)**.** *Intuitively, because a Householder transformation is a reflection, the angular separation is also preserved. However, it is not difficult to directly show that the Householder Transform preserves angular separation. Formally,* $\dfrac{(\boldsymbol{H}\boldsymbol{u})^T(\boldsymbol{H}\boldsymbol{v})}{\|\boldsymbol{H}\boldsymbol{u}\|\|\boldsymbol{H}\boldsymbol{v}\|} = \dfrac{\boldsymbol{u}^T\boldsymbol{v}}{\|\boldsymbol{u}\|\|\boldsymbol{v}\|}$.

*Proof.* Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two arbitrary vectors. Observe that

$$
\begin{aligned}
(\boldsymbol{H}\boldsymbol{u})^T &= (\boldsymbol{u} - 2\boldsymbol{n}\boldsymbol{n}^T\boldsymbol{u})^T = \boldsymbol{u}^T - 2\boldsymbol{u}^T\boldsymbol{n}\boldsymbol{n}^T \\
(\boldsymbol{H}\boldsymbol{v}) &= \boldsymbol{v} - 2\boldsymbol{n}\boldsymbol{n}^T\boldsymbol{v} \\
(\boldsymbol{H}\boldsymbol{u})^T(\boldsymbol{H}\boldsymbol{y}) &= (\boldsymbol{u}^T - 2\boldsymbol{u}^T\boldsymbol{n}\boldsymbol{n}^T)(\boldsymbol{v} - 2\boldsymbol{n}\boldsymbol{n}^T\boldsymbol{v}) \\
&= \boldsymbol{u}^T\boldsymbol{v} - \boldsymbol{u}^T(2\boldsymbol{n}\boldsymbol{n}^T\boldsymbol{v}) - 2\boldsymbol{u}^T\boldsymbol{n}\boldsymbol{n}^T(\boldsymbol{v}) + 4(\boldsymbol{u}^T\boldsymbol{n}\boldsymbol{n}^T\boldsymbol{n}\boldsymbol{n}^T\boldsymbol{v}) \\
&= \boldsymbol{u}^T\boldsymbol{v} - 2\boldsymbol{u}^T\boldsymbol{n}\boldsymbol{n}^T\boldsymbol{v} - 2\boldsymbol{u}^T\boldsymbol{n}\boldsymbol{n}^T\boldsymbol{v} + 4\boldsymbol{u}^T\boldsymbol{n}(1)\boldsymbol{n}^T\boldsymbol{v}
\end{aligned}
$$

$$= \boldsymbol{u}^T \boldsymbol{v}$$

Observing that $\|\boldsymbol{Hu}\|^2 = (\boldsymbol{Hu})^T \boldsymbol{Hu} = \boldsymbol{u}^T \boldsymbol{H}^T \boldsymbol{Hu} = \boldsymbol{u}^T \boldsymbol{u} = \|\boldsymbol{u}\|_2^2$ and $\|\boldsymbol{Hv}\|^2 = (\boldsymbol{Hv})^T \boldsymbol{Hv} = \boldsymbol{v}^T \boldsymbol{H}^T \boldsymbol{Hv} = \boldsymbol{v}^T \boldsymbol{v} = \|\boldsymbol{v}\|_2^2$ completes the proof. □

Note: $\|\boldsymbol{u} \odot \boldsymbol{u}^\perp\|_2^2$ can in some cases be less than 1, and applying a Householder Transform would in those cases result in a larger upper bound on the flipping probability for cases where the angular separation is close to $\pi/2$. However, in most cases, applying this Householder Transform can significantly improve the flipping probability.

## 5.3 Empirical Corroboration of Theorems

Here, in this section, we present experimental results which corroborate our theory developed in Section 5.2.

### 5.3.1 Empirical Validation

We set two orthogonal vectors $\boldsymbol{u}$ and $\boldsymbol{u}^\perp \in \mathbb{R}^d$, with the vectors specially constructed to have the desired $d\|\boldsymbol{u}\|_4^4$ and $d\|\boldsymbol{u} \odot \boldsymbol{u}^\perp\|_2^2$ values. We do this by changing the proportion of non-zero elements in the vector $(s)$ and choosing specific distributions to generate the vector (see appendix B). Table 5.1 shows the definitions we used to define $\boldsymbol{u}$ and $\boldsymbol{t}$ as well as a summary of the values of $d\|\boldsymbol{u}\|_4^4$ and $d\|\boldsymbol{u} \odot \boldsymbol{u}^\perp\|_2^2$ for the various distributions used to define $\boldsymbol{u}$ and $\boldsymbol{u}^\perp$.

We then set $\boldsymbol{v} = \boldsymbol{u}\cos(\theta) + \boldsymbol{u}^\perp \sin(\theta)$. Note, by construction $\boldsymbol{u}$ and $\boldsymbol{v}$ is separated with an angular separation of $\theta$. We then apply $N_p = 10000$ random subspace projection of $k-$subspaces on $\boldsymbol{u}$ and $\boldsymbol{v}$ and empirically measure the proportion of label flipping $f_p = \frac{|(\boldsymbol{Pu})^T(\boldsymbol{Pv})/\boldsymbol{u}^T\boldsymbol{v} <= 0|}{N_p}$, where $|A|$ is the count of the number of elements in $A$. We repeat this for a range of $\theta \in [0, \pi]$ with a step size of $\pi/100$ and plot $f_p$ vs $\theta$ for $k \in [1, 5, 10, 20, 50, 100]$.

Using algorithm C.1, we applied a Householder transform $\boldsymbol{H}$ to reflect $\boldsymbol{u}$ such that every entry of $\boldsymbol{Hu}_i = \pm 1/\sqrt{d}$ (with probability 1/2). We then plotted the flipping probability of $(\boldsymbol{Hu})^T \boldsymbol{P}(\boldsymbol{Hv})$ to give a visual comparison of the improvement to the flipping probability after applying the Householder transformation.

| $\boldsymbol{u}$ | $\boldsymbol{u}^{\perp}$ | $d\Vert\boldsymbol{u}\Vert_4^4$ | $d\Vert\boldsymbol{u}\odot\boldsymbol{u}^{\perp}\Vert_2^2$ |
|---|---|---|---|
| $u_i := \begin{cases} N(0,1) & i \leq s \\ 0 & s < i \leq d \end{cases}$ | $\boldsymbol{u}^{\perp}:= \boldsymbol{v}- <\boldsymbol{v},\boldsymbol{u}> \boldsymbol{u}$<br>with $\boldsymbol{v} := N(0,I)$ | $3d/s$ | 1 |
| $u_i := \begin{cases} -1/\sqrt{s} & w.p.\ 1/2, i \leq s \\ 1/\sqrt{s} & w.p.\ 1/2, i \leq s \\ 0 & s < i \leq d \end{cases}$ | $\boldsymbol{u}^{\perp}:= \boldsymbol{v}- <\boldsymbol{v},\boldsymbol{u}> \boldsymbol{u}$<br>with $\boldsymbol{v} := N(0,I)$ | $d/s$ | 1 |
| $u_i := \begin{cases} -1/\sqrt{s} & w.p.\ 1/2, i \leq s \\ 1/\sqrt{s} & w.p.\ 1/2, i \leq s \\ 0 & s < i \leq d \end{cases}$ | $u_i^{\perp} := \begin{cases} -u_i & i \leq s/2 \\ u_i & s/2 < i \leq s \\ 0 & s < i \leq d \end{cases}$ | $d/s$ | $d/s$ |
| $u_i := \begin{cases} -1/\sqrt{s} & w.p.\ 1/2, i \leq s \\ 1/\sqrt{s} & w.p.\ 1/2, i \leq s \\ 0 & s < i \leq d \end{cases}$ | $u_i^{\perp} := \begin{cases} -qu_i & i \leq s/2 \\ qu_i & s/2 < i \leq s \\ \sqrt{\frac{1-q^2}{d-s}} & s < i \leq d \end{cases}$ | $d/s$ | $dq^2/s$ |
| $u_i := \begin{cases} -1/\sqrt{s} & w.p.\ 1/2, i \leq s \\ 1/\sqrt{s} & w.p.\ 1/2, i \leq s \\ 0 & s < i \leq d \end{cases}$ | $u_i^{\perp} := \begin{cases} 0 & i \leq s \\ \frac{1}{\sqrt{d-s}} & s < i \leq d \end{cases}$ | $d/s$ | 0 |

**Table 5.1:** *Summary of the definition of the vectors $\boldsymbol{u}$ and $\boldsymbol{u}^{\perp}$ and the corresponding values of $d\Vert\boldsymbol{u}\Vert_4^4$ and $d\Vert\boldsymbol{u}\odot\boldsymbol{u}^{\perp}\Vert_2^2$*

These results are plotted in Figures 5.1 through 5.8, with the horizontal axes the angular separation in $\pi$ radians and the vertical axes, is the proportion of label flipping (flipping probability). We include the additional figures in the appendix, namely Figures D.1 through D.15. The solid purple lines in the figures are the theoretical upper-bound of flip probability as stated by Theorem 5.6. The blue, red and yellow plots are the empirical flipping probability for Gaussian RP projected vectors, RS projected vectors and RS projected vectors with Householder 'densification' applied respectively.

### 5.3.2 Discussion of the Empirical Results

We can see in the Figures 5.1 through 5.8 that our theoretical bound captures the shape of the empirical flip probability accurately, albeit with an offset especially for small values of $k$ ($k < 4d\|\boldsymbol{u}\|_4^4$) (see Figures 5.3, 5.6, 5.8 ). This offset is dependent on $d\|\boldsymbol{u}\|_4^4$ and captures probability that the random subspace projection picks a feature values that is very small in comparison to the rest of the entries.

To see why this is so, we consider this extreme but straight forward example with these three vectors, $\boldsymbol{u} = [\sqrt{1-(d-1)\epsilon^2}, \epsilon, \ldots, \epsilon]$; $\boldsymbol{u}' = [\sqrt{1-(d-1)\epsilon^2}, -\epsilon, \ldots, -\epsilon]$; and $\boldsymbol{v} = [\cos\theta, \sqrt{\frac{1}{d-1}}\sin\theta, ..., \sqrt{\frac{1}{d-1}}\sin\theta]$ each with an arbitrary small $\epsilon$. Observe that both $\boldsymbol{u}, \boldsymbol{v}$ and $\boldsymbol{u}', \boldsymbol{v}$ has an angular separation $\theta$. We can also see that both $d\|\boldsymbol{u}\|_4^4 = d\|\boldsymbol{u}'\|_4^4 = d$. However, if we were to combinatorially calculate the flipping probabilities of $\boldsymbol{u}^T\boldsymbol{P}\boldsymbol{v}$ and $\boldsymbol{u}'^T\boldsymbol{P}\boldsymbol{v}$ for $\theta \in [0, \pi/2)$, we have $\Pr\left\{\boldsymbol{u}^T\boldsymbol{P}\boldsymbol{v} < 0\right\} = 0$ (since every entry of $u_i v_i > 0$) and $\Pr\left\{\boldsymbol{u}'^T\boldsymbol{P}\boldsymbol{v} < 0\right\} = 1-k/d < e^{-k/d}$ (since only the first entry of $u_i v_i > 0$ and every other entry $< 0$).

We also note that our theorem gives an upper bound on a flipping probability of 1 when the angular separation approaches $\pi/2$. Again, this is not unexpected. Consider these two pairs of vector $\boldsymbol{u} = [1, \epsilon, ..., \epsilon], \boldsymbol{v} = [\epsilon, ..., \epsilon, 1]$ and $\boldsymbol{u}' = [1, -\epsilon, ..., -\epsilon], \boldsymbol{v}' = [3\epsilon, \epsilon, ..., \epsilon, 1]$. Both $\boldsymbol{u}\boldsymbol{v}$ and $\boldsymbol{u}'\boldsymbol{v}'$ have an angular separation of $\cos^{-1} 2\epsilon$ but has entries that have signs flipped almost everywhere except for the first entry. As in the previous example, we can see that the flipping probability for the second pair is $1 - k/d$ using combinatorial techniques.

We would like to note that in Figure 5.3, there is a slight skewness the flipping probability of the random subspace projected vectors. This comes from $\|\boldsymbol{u}^3 \odot \boldsymbol{u}^\perp\|_1$

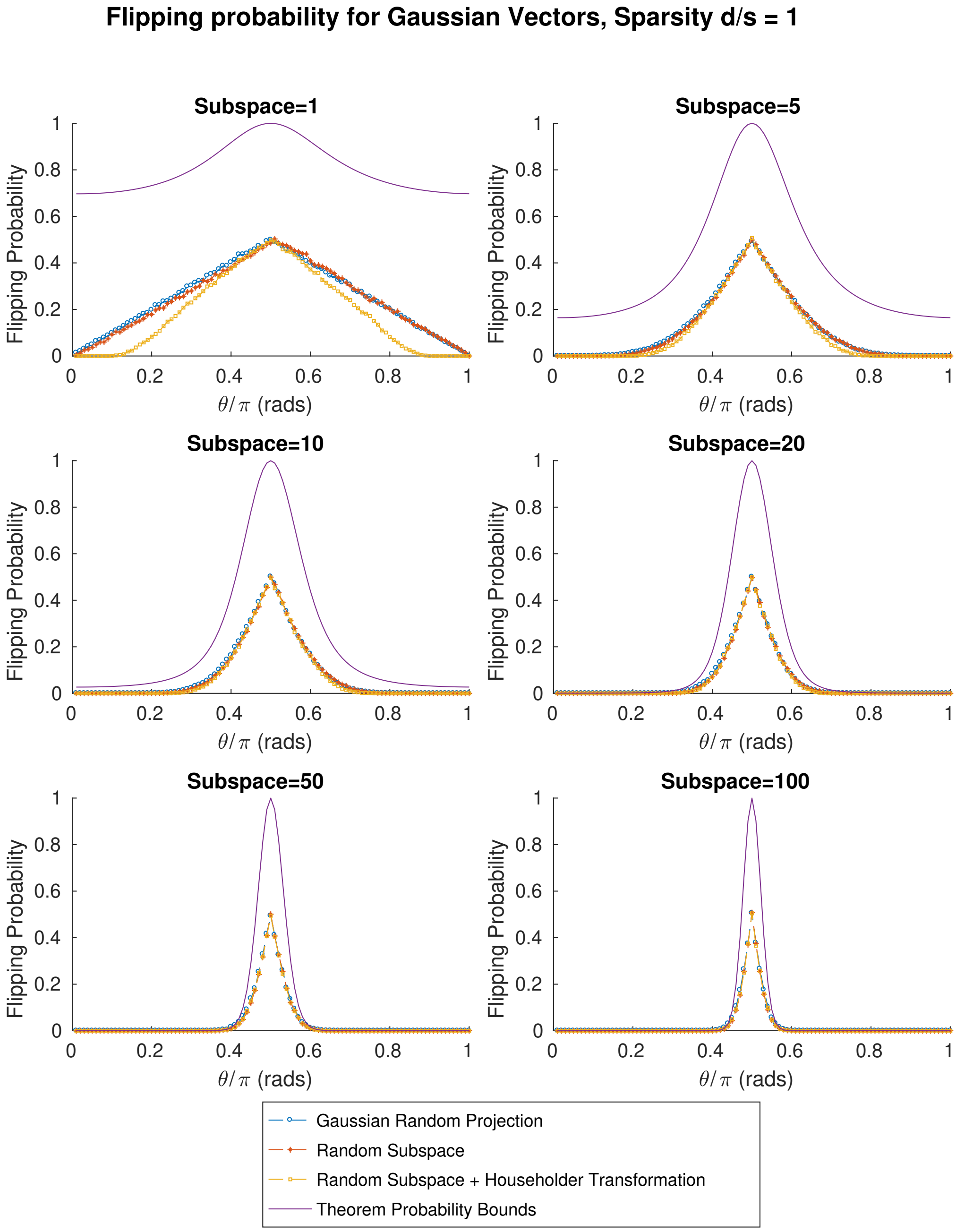

**Figure 5.1:** *Flipping probability vs angular separation for Gaussian vectors, (first row in Table 5.1), with sparsity $s = 1$ for projection dimension $k \in \{1, 5, 10, 20, 50, 100\}$ and dimensionality $d = 1000$. Observe that our theory upper bounds the flipping probability.*

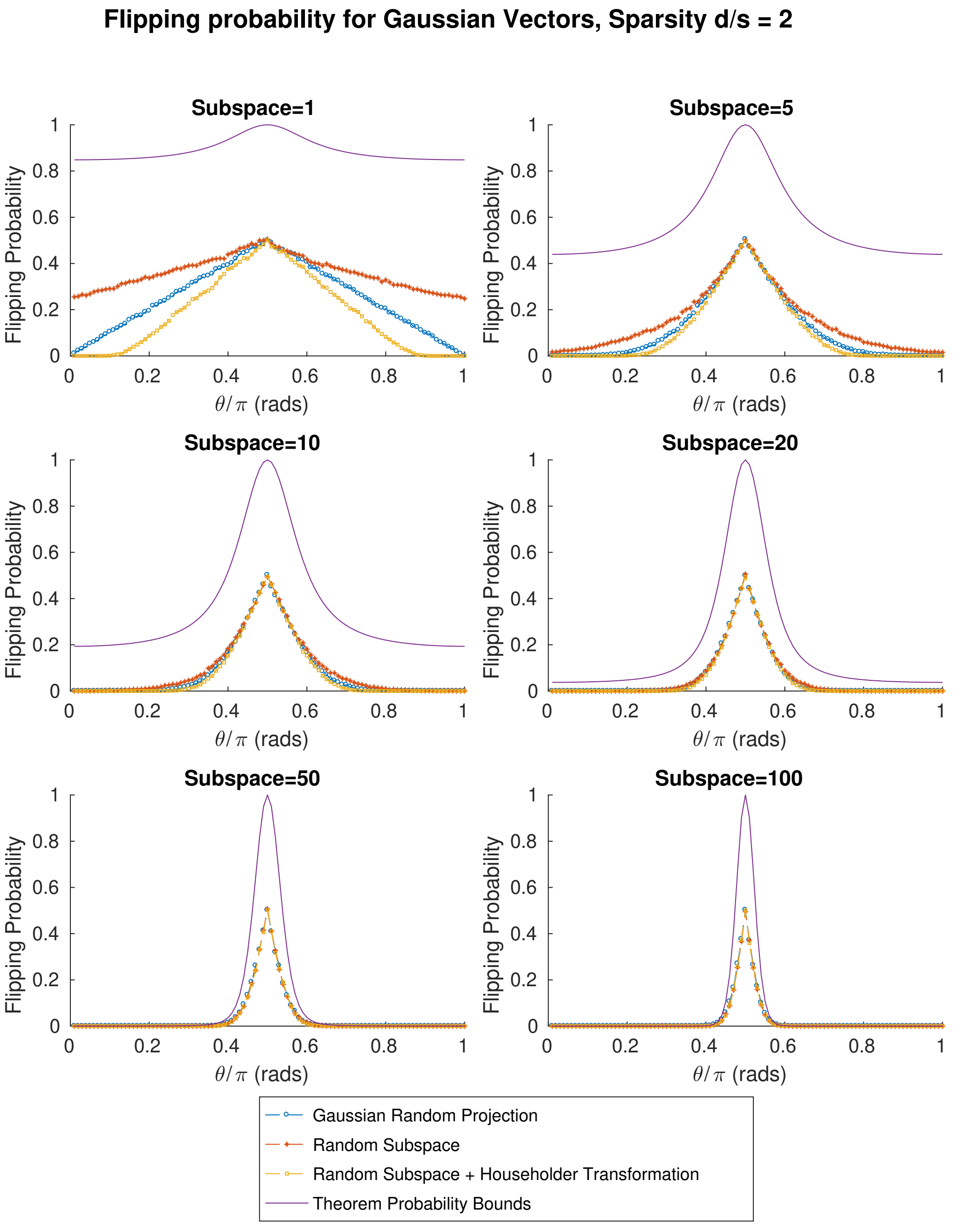

**Figure 5.2:** *Flipping probability vs angular separation for Gaussian vectors, (first row in Table 5.1) with sparsity $s = 2$ for projection dimension $k \in \{1, 5, 10, 20, 50, 100\}$ and dimensionality $d = 1000$.*

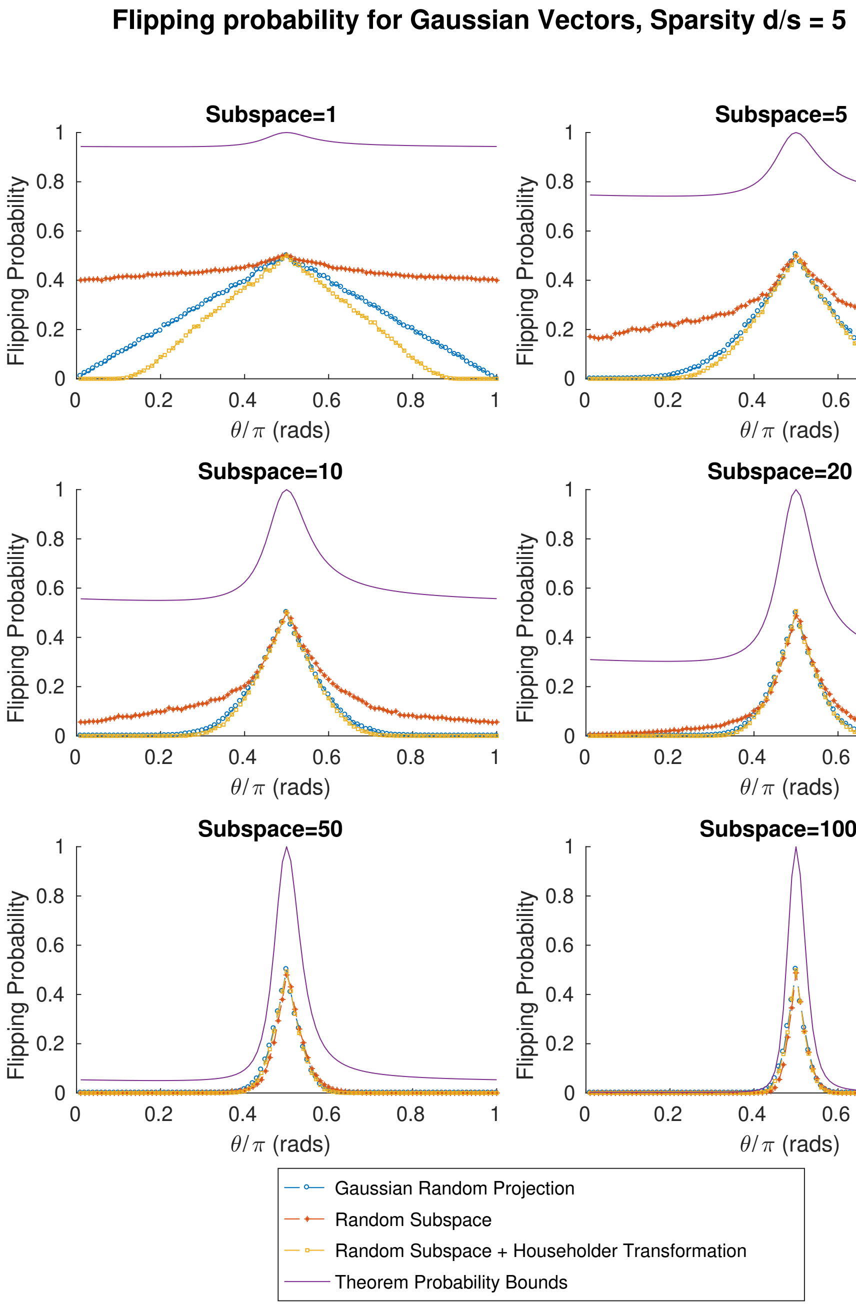

**Figure 5.3:** *Flipping probability vs angular separation for Gaussian vectors, (first row in Table 5.1) with sparsity $s = 5$ for projection dimension $k \in \{1, 5, 10, 20, 50, 100\}$ and dimensionality $d = 1000$. Observe that the non-symmetric behaviour of the empirical flipping probability is predicted by our theorem.*

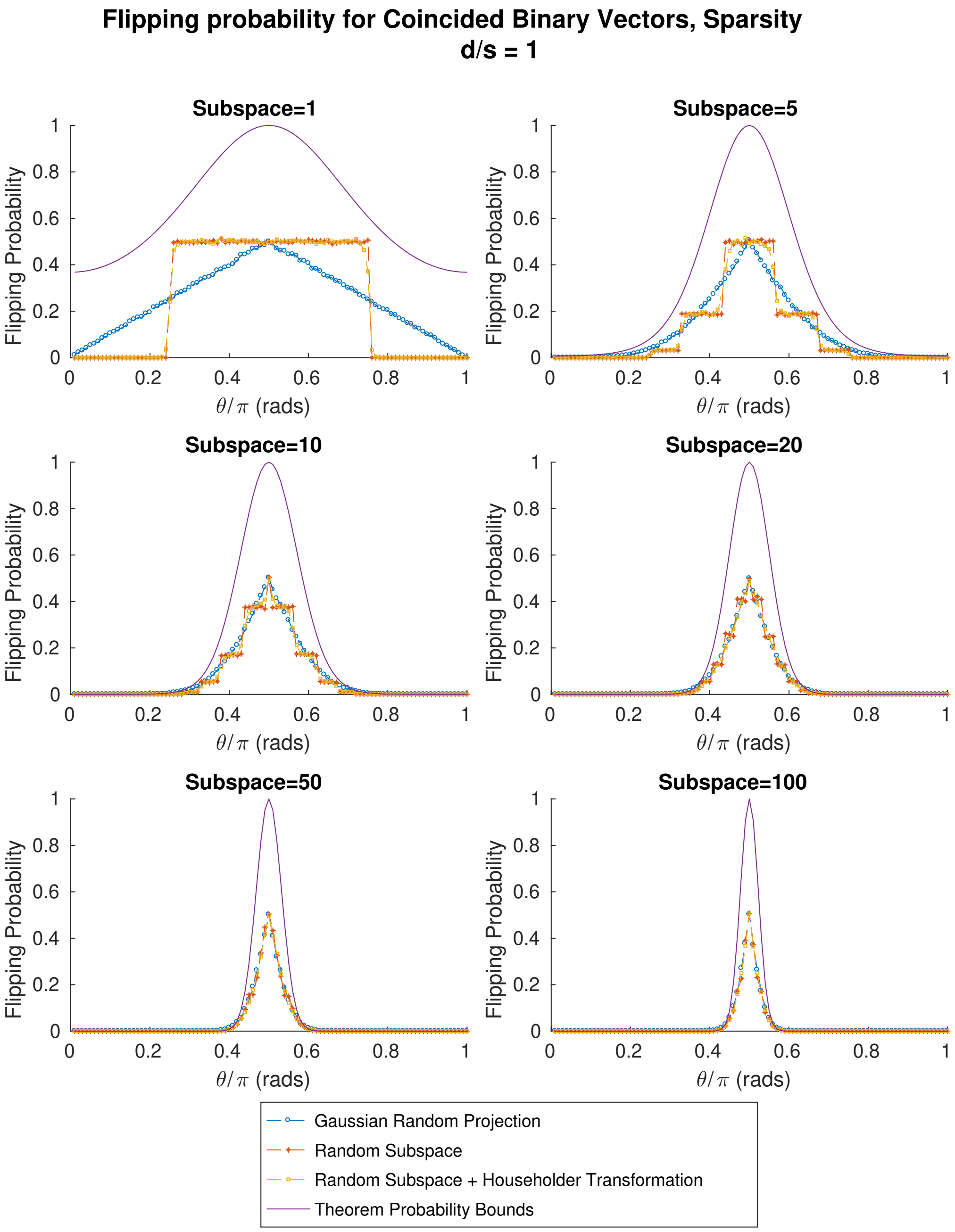

**Figure 5.4:** *Flipping probability vs angular separation for two binary vector that coincides in every coordinate, (row three in Table 5.1) with sparsity $s = 1$ for projection dimension $k \in \{1, 5, 10, 20, 50, 100\}$ and dimensionality $d = 1000$.*

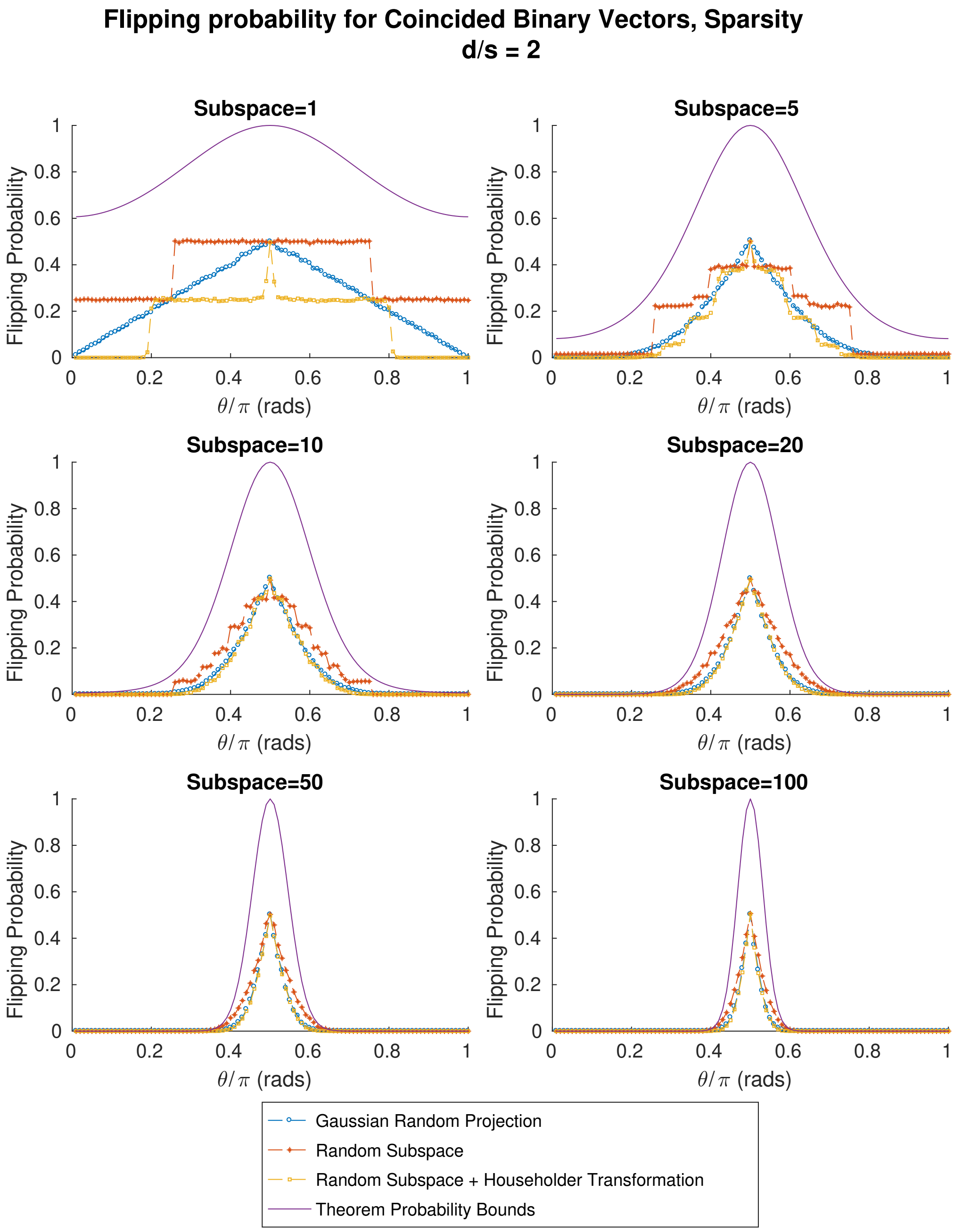

**Figure 5.5:** *Flipping probability vs angular separation for two binary vector that coincides in every coordinate, (row three in Table 5.1) with sparsity $s = 2$ for projection dimension $k \in \{1, 5, 10, 20, 50, 100\}$ and dimensionality $d = 1000$.*

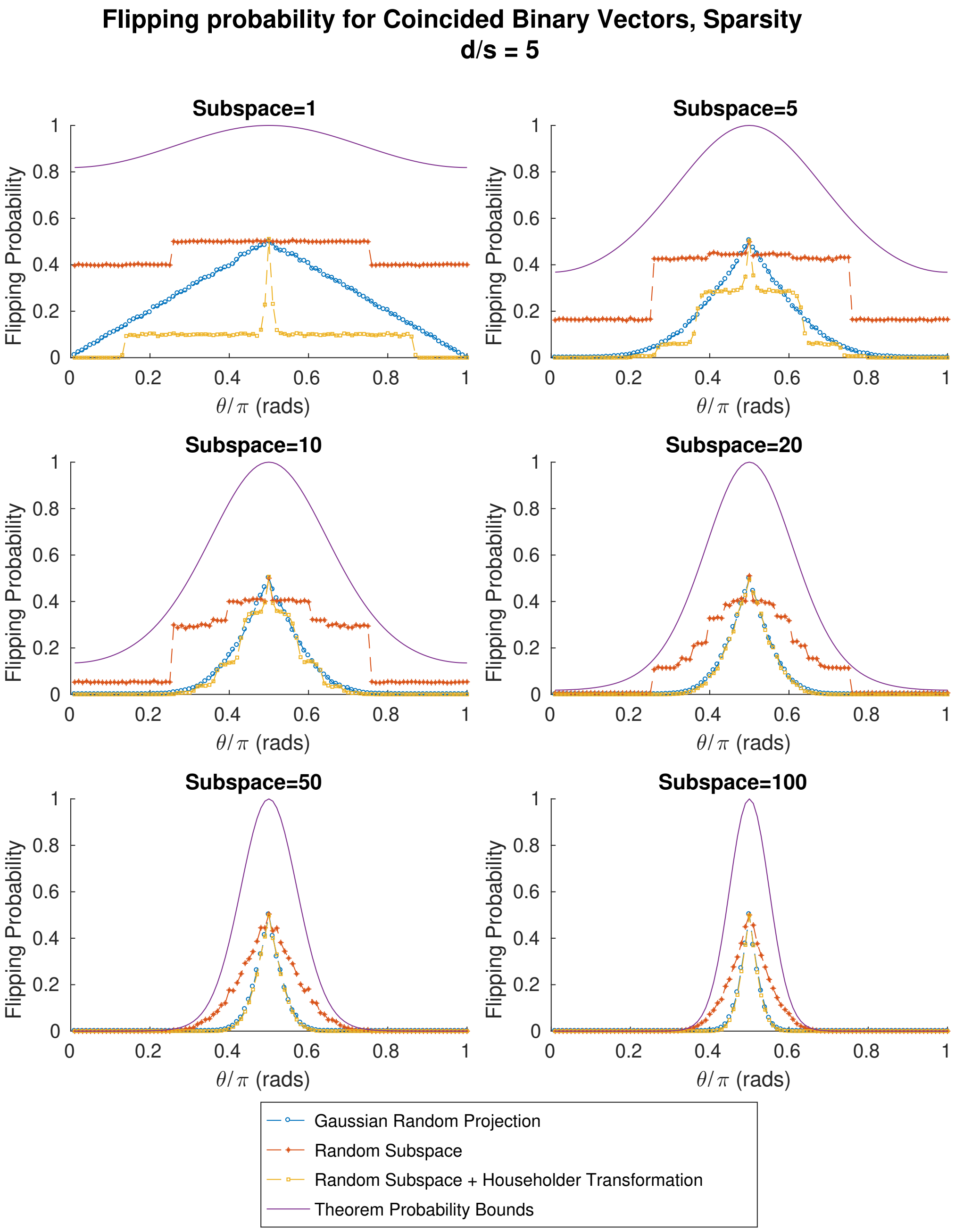

**Figure 5.6:** *Flipping probability vs angular separation for two binary vector that coincides in every coordinate, (row three in Table 5.1) with sparsity $s = 5$ for projection dimension $k \in \{1, 5, 10, 20, 50, 100\}$ and dimensionality $d = 1000$.*

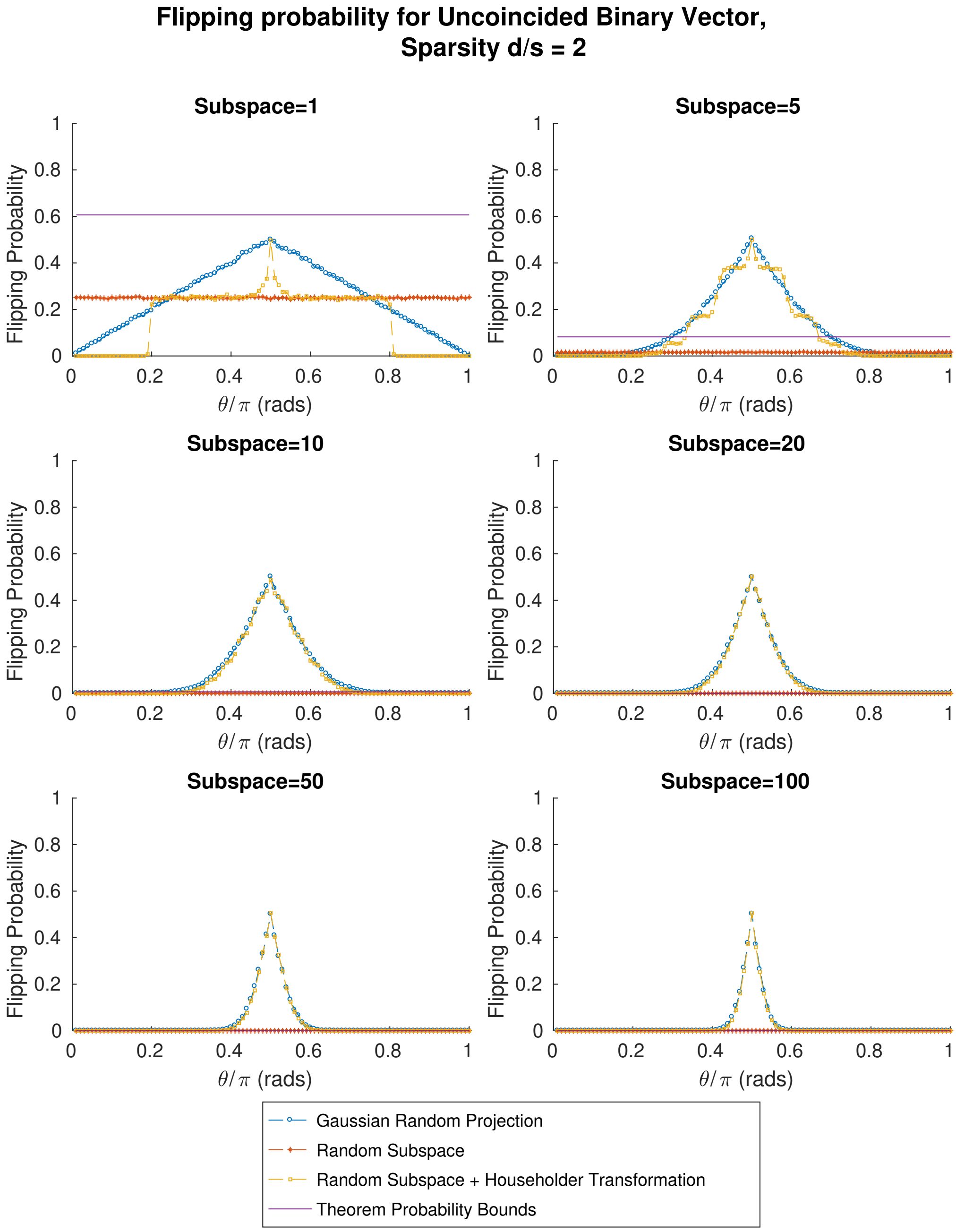

**Figure 5.7:** *Flipping Probability vs Angular Separation of two binary vectors such that the two vectors do not coincide, (row five in Table 5.1) with sparsity $s = 2$ for projection dimension $k \in \{1, 5, 10, 20, 50, 100\}$ and dimensionality $d = 1000$. Observe that applying Householder transform increases the flipping probability.*

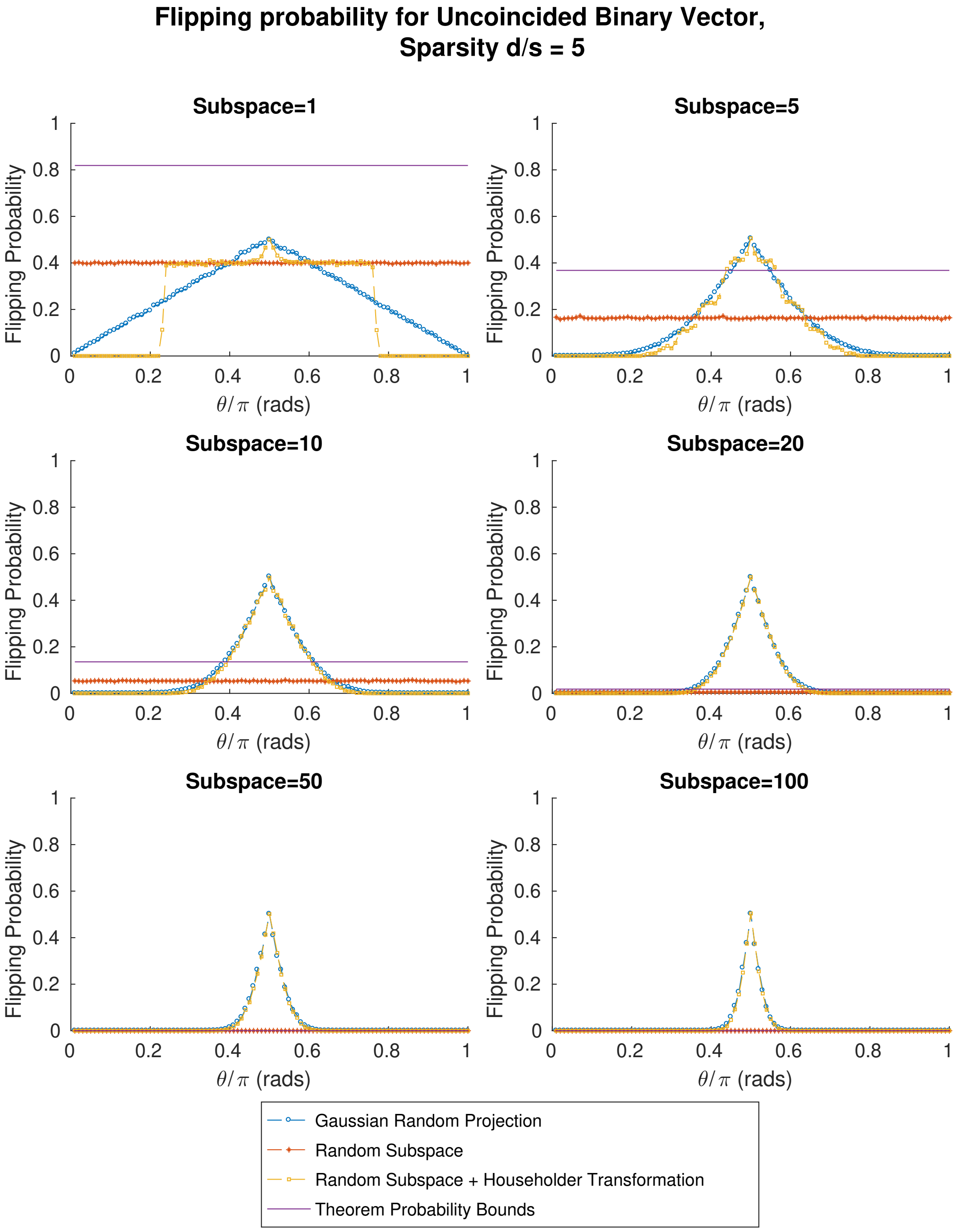

**Figure 5.8:** *Flipping Probability vs Angular Separation of two binary vectors such that the two vectors do not coincide, (row five in Table 5.1) with sparsity $s = 5$ for projection dimension $k \in \{1, 5, 10, 20, 50, 100\}$ and dimensionality $d = 1000$. Observe that applying Householder transform increases the flipping probability.*

being large enough to be significant and the non-symmetric nature of the flipping probability was also captured by our theorem.

We also see from our figures, that the Householder transformation can be used to improve the flipping probabilities except for the cases when $d\|\boldsymbol{u}\|_4 = 1$ , in which case $\boldsymbol{u}$ is already dense, and there would not be any improvement gained by using the Householder transformation) or in cases when $d\|\boldsymbol{u} \odot \boldsymbol{u}^{\perp}\|_2 < 1$ (see Figures 5.7 and 5.8). $\boldsymbol{u} \odot \boldsymbol{v}$ is already fairly regular and by applying Householder Transform makes the vectors less regular, and increases the flipping probability but not much more than using Gaussian RP projection. In most cases, applying the Householder Transform would improve the flipping probability especially for a pair of sparse vectors.

## 5.4 Implication for Classification Ensembles

Our results above suggest that an ensemble of randomly projected classifiers (in this case random subspace projection), can be used as an ensemble to recover the Bayes' classifier. To see why this is so, consider a linear classifier with decision boundary described by $\boldsymbol{h}$. Let $\boldsymbol{h}_i := \boldsymbol{h}\boldsymbol{P}_i$ be a random projection of $\boldsymbol{h}$. Observe that the errors of projected classifier can be decomposed and an upper bound of the error is

$$\begin{aligned}
\mathrm{E}[\mathbf{1}(\boldsymbol{h}_i^T\boldsymbol{x}) \neq y] &= \mathrm{E}[\mathbf{1}(\boldsymbol{h}^T\boldsymbol{x} \neq y)] + \mathrm{E}[\mathbf{1}(\boldsymbol{h}_i^T\boldsymbol{x} \neq \boldsymbol{h}^T\boldsymbol{x} \cap \boldsymbol{h}^T\boldsymbol{x} = y)] \\
&\quad - \mathrm{E}[\mathbf{1}(\boldsymbol{h}_i^T\boldsymbol{x} \neq \boldsymbol{h}^T\boldsymbol{x} \cap \boldsymbol{h}^T\boldsymbol{x} \neq y)] \\
&\leq \mathrm{E}[\mathbf{1}(\boldsymbol{h}^T\boldsymbol{x} \neq y)] + \mathrm{E}[\mathbf{1}((\boldsymbol{h}_i^T\boldsymbol{x} \neq \boldsymbol{h}^T\boldsymbol{x}) \cap (\boldsymbol{h}^T\boldsymbol{x} = y))] \\
&\quad + \mathrm{E}[\mathbf{1}((\boldsymbol{h}_i^T\boldsymbol{x} \neq \boldsymbol{h}^T\boldsymbol{x}) \cap (\boldsymbol{h}^T\boldsymbol{x} \neq y))] \\
&= \mathrm{E}[\mathbf{1}(\boldsymbol{h}^T\boldsymbol{x} \neq y)] + \mathrm{E}[\mathbf{1}(\boldsymbol{h}_i^T\boldsymbol{x} \neq \boldsymbol{h}^T\boldsymbol{x})]
\end{aligned}$$

Moreover, observe that $\mathrm{E}[\mathbf{1}(\boldsymbol{h}_i^T\boldsymbol{x} \neq \boldsymbol{h}^T\boldsymbol{x})]$ is the flipping probability and that the flipping probability is independent of the other instances of the random subspace projections (i.e. $\mathrm{E}[\mathbf{1}(\boldsymbol{h}_i^T\boldsymbol{x} \neq \boldsymbol{h}^T\boldsymbol{x})]$ is independent of $\mathrm{E}[\mathbf{1}(\boldsymbol{h}_j^T\boldsymbol{x} \neq \boldsymbol{h}^T\boldsymbol{x})]$ for all $i \neq j$).

Therefore, by Condorcet's Jury Theorem, this implies that as the ensemble size $N$ tends to infinity, the majority vote accuracy of the classification ensemble will tend to the accuracy of the Bayes' classifier, assuming of course, that the average

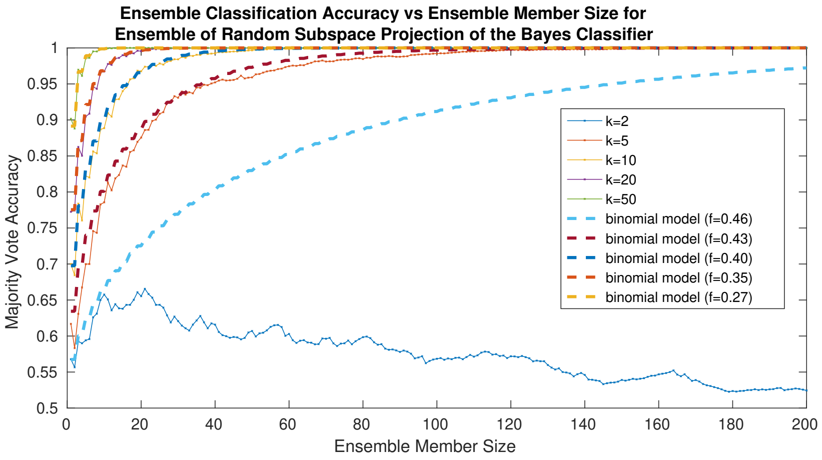

**Figure 5.9:** *Majority vote accuracy for classification ensemble of Random Subspace projection of the Bayes' classifier with dimensionality $d = 1000$ and sparsity $d/s = 2$. Dashed lines are the majority vote accuracy as predicted by a binomial model with the average flipping as the parameter of the binomial model, the flipping probability $f$ was determined through empirical simulation. Observe here that the ensemble failed to give a consistent model for $k = 2$. While the flipping probability in the legends is less than* 0.5*, the flipping probability values in the legend excluded projection angles that are exactly $\pi/2$. Empirical flipping probability for $k = 2$ is* 0.55*.*

flipping probability is less than 0.5.

$$\lim_{N\to\infty} \mathrm{E}[\sum_{i=1}^{N} \mathbf{1}(\boldsymbol{h}_i^T \boldsymbol{x} \neq y)] = \mathrm{E}[\mathbf{1}(\boldsymbol{h}^T \boldsymbol{x} \neq y)]$$

Figure 5.9 shows the majority vote accuracy of an ensemble classifiers where the member classifier is generated by applying random subspace projection on $\boldsymbol{h}$ (i.e. $\boldsymbol{h}_i := \boldsymbol{h}\boldsymbol{P}_i$). The dashed lines are the majority vote accuracy as predicted by a binomial model with the average flipping as the parameter of the binomial model. Observe that the accuracy of the ensemble follows a binomial distribution with the parameter of the binomial distribution determined by the flipping probability.

However, because learning algorithms generate a hypothesis based on a finite set of training examples, and reasonable learning algorithms learn hypotheses that maximizes the margins, the errors of the hypothesis $\boldsymbol{h}_i$ generated by the learning algorithm from randomly projecting training data with $\boldsymbol{P}_i$ is not independent. In the next chapter, we will look at modelling the majority vote classification ensemble when the errors of the member classifiers are not independent by leveraging on results from the social sciences and economics and using the Polya-Eggenberger distribution.

## 5.5 Conclusion and Summary

In this chapter, we have derived the sub-Gaussian norm of a vector representing the random subspace projected data. We applied the sub-Gaussian norms to the theorems in Kabán and Durrant (2017) giving us the flipping probability and generalization errors of random subspace projected classifiers.

We derived the data-dependent flipping probability and empirically showed how our bounds capture the empirical flipping probabilities of random subspace projected vectors. We provided an analogue to the regularity constant and discussed the factors affecting the flipping probabilities.

We also demonstrated how using Householder transformations could be used to improve the flipping probability of the vectors.

One idea to improve the flipping probability for a general random subspace projected classifier is to use a Householder Transforms to reflect the normal vector of hyperplane representing the Bayes optimal classifier such that it is dense.

We also discussed the implications of the flipping probability for classification ensembles and show the intuition of how our theory on flipping probability shows how we can recover the Bayes' classifier by considering the independence in the errors of the randomly projected classifiers. We also discussed the limitations of that intuition and why in practice, the errors of classifiers generated by randomly subspace projected data is not independent.

In the next chapter, we will look at results from the social sciences and show how we can model the majority vote accuracy of the ensemble when the classifiers are correlated using a Polya-Eggenberger distribution. We will show how the Sneath and Sokal (1963) diversity measure estimates the parameters of the distribution and discuss the implications of the model.

# 6

# Ensembles of Random Subspace Classifiers

**Summary** In the previous chapter, we discussed the intuition of how an ensemble of randomly subspace projected classifier can recover the Bayes' classifier by observing that the flipping probability of a random subspace projected classifier is independent of the other randomly projected classifiers. However, because the learning algorithm learns the classifier on a finite set training set, the flipping errors in the individual classifiers are not independent.

In this chapter, we investigate the accuracy of a majority vote classification ensemble by modelling the accuracy with a Polya-Eggenberger distribution as described by Ladha (1995) and Berg (1993). We will show we can use the Sneath and Sokal (1963) $\rho$ diversity measure to estimate the dispersion parameter $\psi$ of the Polya-Eggenberger distribution. We discuss the suitability of this model and we decompose the model using "good" and "bad" diversity error decomposition as defined by Brown and Kuncheva (2010). We also evaluate other proposed methods of estimating diversity including the methods proposed by Ladha (1995) and Berg (1993).

We also discuss various combination schemes such as sum rule, and in our empirical exploration we will try to reconcile the contradictory findings of Kuncheva and Rodríguez (2014) to Schapire (1990); Blum (1997) and we explore the intuition on why RS ensembles can be considered as a regularization of the original high-dimension problem.

We empirically compare the ensemble accuracy on different ensemble combination schemes. We also compare the Polya-Eggenberger model to our empirical results

and show that our choice of diversity measure is a reasonable estimate for the model dispersion parameter. Finally, we compare our theoretical to empirical experience using findings on high-dimensional data from the NIPS 2003 Feature Selection Challenge (Guyon, 2003).

## 6.1 Background

Empirical results have shown that ensemble classifiers are typically superior in terms of accuracy and robustness versus individual learners. It is generally accepted that the accuracy of an ensemble classifier tends to increase with increasing ensemble size,and with the accuracy of the individual classifiers and diversity between the ensemble members. However, the actual relationship between these aspects is mostly unknown.

## 6.2 Majority Voting

As we may recall from section 2.1.2, the choice of the combination method in the combination scheme for an ensemble can significantly affect the overall accuracy of the ensemble learner. Of the many combination methods, the most studied and commonly used combination scheme is the majority vote. In a majority vote ensemble, each classifier chooses a class label, and the class label chosen by the greatest number of classifiers is selected as the output of the ensemble. Some literature distinguishes between plurality vote and majority vote in this context; however, for a two-class classification, these two combination schemes are mathematically identical.

In the early literature on majority vote ensemble classifiers, the accuracy models for majority vote are based on the binomial model which assume independence of votes and does not take into account the diversity of the classifiers in the ensemble (Lam and Suen, 1997; Whitaker and Kuncheva, 2003; Kuncheva et al., 2003). This is in spite of the empirical evidence showing that diversity is important to the ensemble accuracy and this makes it challenging to optimize the accuracy-diversity trade-off for the ensemble using the binomial model since the assumption of independent errors is typically false. To put this another way, such theory is weak in the sense that it ignores aspects of the problem that are known empirically to be important.

Meanwhile, results from the field of Social Sciences namely those by Ladha (1995) and Berg (1993), propose that the accuracy of a majority vote voting system can be modelled using a Polya-Eggenberger distribution (which is a generalization of the well-known Beta-Binomial model, but allows for a limited range of negative valued shape parameters). While this model is still fairly restrictive — in particular the model assumes identical competencies in the voters (i.e. identical probabilities that the votes are correct) — our empirical results in section 6.5.1 show that the assumption of identical classifier competencies is not too unrealistic and the accuracy of a majority vote classification can be modelled quite accurately by a Polya-Eggenberger distribution even though the assumption of identical classifier competencies is not often met in practice.

### 6.2.1 Polya-Eggenberger Distribution

The Polya-Eggenberger model is a distribution describing the expected number of successes in $N$ trials drawing from the Polya urn model. In the basic Polya urn model, we have $a$ black balls (successes) and $b$ white balls (failure) in an urn. One ball is drawn randomly from the urn and the colour of the ball is observed. The ball is returned to the urn, and $s$ balls of the same colour are also added. This makes it more likely that an observation that happened previously will be repeated when $s$ is positive, analogous to two correlated classifiers being more likely to vote similarly, and less likely when $s$ is negative. The Polya-Eggenberger model generalizes the distribution to allow non-integer $a$ and $b$, and negative-valued $s$ (Feller, 2008; Sen and Mishra, 1996).

**Definition 6.1** (Polya-Eggenberger Distribution (Sen and Mishra, 1996))**.** *Let $N$ be the number of trials in a Polya urn model, let the initial number of black balls be $a$ and the initial number of white balls be $b$. Let the number of additional balls to be added following an observation (of black or white) be $s$. Define $S_N$ to be the number of black balls drawn after $N$ trials. Define $p := \frac{a}{a+b}$ and $\psi := \frac{s}{a+b}$.*

*Then $S_N$ follows a Polya-Eggenberger distribution with the following definition*

*Case 1: $\psi \geq -\frac{1}{N}, \psi \neq 0$*

$$Pr\{S_N = k\} = \frac{\binom{-\frac{p}{\psi}}{k}\binom{-\frac{1-p}{\psi}}{n-k}}{\binom{-\frac{1}{\psi}}{n}}$$

*Case 2:* $\psi = 0$

$$Pr\{S_N = k\} = \binom{n}{k} p^k (1-p)^{n-k}$$

*With* $\binom{x}{y}$ *defined for any real* $x$ *and integer* $y$ *as*

$$\binom{x}{y} := \frac{(x)(x-1)\dots(x-y+1)}{y!}$$

*Note that* $\binom{x}{y}$ *can also be written as* $\frac{\Gamma(x+1)}{\Gamma(x-y+1)\Gamma(y+1)}$ *when* $x \geq y$. $\binom{x}{y} = (-1)^y \binom{-x+y-1}{y} = (-1^y)\frac{\Gamma(y-x)}{\Gamma(-x)\Gamma(y+1)}$ *when* $x < y$ *where* $\Gamma(x)$ *is the Gamma function.*

The parameter that $\psi$ can be interpreted as the increased likelihood that an observation would be the same colour as the previous observation. In other words, if $p_i$ is the probability that the $i$-th observation is a success, then $p_{i+1} = \frac{p_i+\psi}{1+\psi}$ if $p_i$ was a success and $p_{i+1} = \frac{p_i}{1+\psi}$ otherwise.

Also note that for $\psi > 0$ this distribution can also be written as a beta-binomial distribution with $\alpha = \frac{p}{\psi}$ and $\beta = \frac{1-p}{\psi}$ However, when $\psi \leq 0$, the values for $\alpha$ and $\beta$ are invalid for such a model and the beta-binomial distribution is undefined. Also, note that $\psi$ has to be greater or equal than $-\frac{1}{N}$ otherwise it implies that a negative number of balls is drawn from the Polya-Urn, violating the physical property of the model — in fact, when $\psi = -\frac{1}{N}$ exactly the model is equivalent to a hyper-geometric distribution (sampling without replacement), and when $\psi = 0$ it is equivalent to a binomial distribution.

In our approach the Polya-Eggenberger model says that the number of classifiers correctly classifying a given example follows a Polya Urn model and therefore the in a majority voting system of $N$ classifiers, the number of ensemble members giving the correct vote can then be given as estimated by :

Case 1: For odd $N$, $\sum_{i=(N+1)/2}^{N} \sum P(S_N = i)$

Case 2: For even $N$, $\sum_{i=(N/2)+1}^{N} \sum P(S_N = i) + \frac{1}{2}P(S_N = N/2)$

This distribution has been studied in Sen and Mishra (1996); Feller (2008); Johnson and Kotz (1977) and its moments are as follows:

- $\mathrm{E}[S_N] = N\frac{a}{a+b} = Np$
- $\mathrm{Var}[S_N] = N\frac{a}{a+b} + \frac{(N^2-N)a(a+s)}{(a+b)(a+b+s)} = \frac{Np(1-p)(N\psi+1)}{1+\psi}$
- $MGF[S_N] =_2 F_1(-N, a/s; (a+b)/s; 1-e^t) =_2 F_1(-N, \frac{p}{\psi}; \frac{1}{\psi}; 1-e^t)$

where $MGF[S_N]$ is the moment generating function for $S_N$, and ${}_2F_1(\boldsymbol{a}; \boldsymbol{b}; c)$ is the ordinary hypergeometric function.

The cumulative distribution function for $S_N$ is:

$$\Pr\{S_N \leq k\} = \begin{cases} 0, & \text{for } k < 0 \\ \binom{n}{k} \frac{\Gamma(k+\frac{p}{\psi})\Gamma(n-k+\frac{1-p}{\psi})\Gamma(\frac{1}{\psi})}{\Gamma(n+\frac{1}{\psi})\Gamma(\frac{p}{\psi})\Gamma(\frac{1-p}{\psi})} {}_3F_2(\boldsymbol{a}; \boldsymbol{b}; 1), & \text{for } 0 \leq k < N \\ 1, & \text{for } k \geq N \end{cases}$$

with $\boldsymbol{a} = (1, -k, N-k+\frac{1-p}{\psi})$ and $\boldsymbol{b} = (N-k-1, 1-k-\frac{p}{\psi})$ and ${}_3F_2(\boldsymbol{a}; \boldsymbol{b}; c)$ is the generalized hypergeometric function (Weisstein, 2002).

For convenience, we will refer to the distribution defined in Theorem 6.1 as $\mathrm{PE}(N, p, \psi)$ where $N$ is the number of trials (or ensemble size), $p = \frac{a}{a+b}$ with $a$ and $b$ the black and white balls in the Polya urn (classifier voting correctly or incorrectly), and $\psi = \frac{s}{a+b}$ with $s$ the number of additional balls added or removed after every trial which would be estimated with the $\rho$ diversity measure of Sneath and Sokal (1963).

### 6.2.2 Correlation and Diversity Measures

Here, we focus on a particular diversity measure, namely the average diversity measure $\rho$ of Sneath and Sokal (1963). We first show that this diversity measure corresponds to the parameter $\psi$ in the definition of the Polya-Eggenberger distribution when the classifiers each have the same accuracy.

We begin by defining $\hat{P}_{ij}$ as the observed proportion of training observations both classifier $i$ and $j$ classified correctly, and $\hat{P}_i$ and $\hat{P}_j$ as the observed proportion of training observations that classifier $i$ and classifier $j$ classified correctly, respectively.

We can then rewrite the $2 \times 2$ contingency table for the pair of classifiers $D_i$ and $D_j$ (Table 6.1) in terms of the $\hat{P}_i$, $\hat{P}_j$ and $\hat{P}_{ij}$.

| | $D_j$ Correct | $D_j$ Wrong |
|---|---|---|
| $D_i$ Correct | $\hat{P}_{ij}$ | $\hat{P}_i - \hat{P}_{ij}$ |
| $D_i$ Wrong | $\hat{P}_j - \hat{P}_{ij}$ | $1 - \hat{P}_i - \hat{P}_j + \hat{P}_{ij}$ |

**Table 6.1:** $2 \times 2$ *contingency table for the classifiers* $D_i$ *and* $D_j$

Recall from Section 2.1.3 $\rho_{i,j}$ is defined as

$$\hat{\rho}_{ij} := \frac{N_{11}N_{00} - N_{01}N_{10}}{\sqrt{(N_{11}+N_{10})(N_{01}+N_{00})(N_{11}+N_{01})(N_{10}+N_{00})}}$$

Suppose that $\forall k \in [1, N]$, $\hat{P}_k \neq 0$,

$$\begin{aligned}\hat{\rho}_{ij} &= \frac{\hat{P}_{ij}(1 - \hat{P}_i - \hat{P}_j + \hat{P}_{ij}) - (\hat{P}_i - \hat{P}_{ij})(\hat{P}_j - \hat{P}_{ij}}{\sqrt{\hat{P}_i\hat{P}_j(1 - \hat{P}_i)(1 - \hat{P}_j)}} \\ &= \frac{\hat{P}_{ij} - \hat{P}_i\hat{P}_j}{\sqrt{\hat{P}_i\hat{P}_j(1 - \hat{P}_i)(1 - \hat{P}_j)}}\end{aligned}$$

Note that $\mathrm{E}[\hat{\rho}_{ij}]$ is of the form $\frac{\mathrm{E}[XY] - \mathrm{E}[X]\mathrm{E}[Y]]}{\sqrt{\mathrm{Var}[X]\mathrm{Var}[Y]}}$, which is the functional definition of the correlation between X and Y. Therefore, $\hat{\rho}_{ij}$ is a sample estimate of the correlation between the outcomes of classifier $i$ and $j$.

Now, let $\hat{r}$ be the average of the $\hat{\rho}_{ij}$ over all pairs $i \neq j$ in the ensemble,

$$\begin{aligned}\hat{r} &:= \frac{1}{N}\frac{1}{N-1}\sum_{i=1}^{N}\sum_{j\neq i}^{N}\frac{\hat{P}_{ij} - \hat{P}_i\hat{P}_j}{\sqrt{\hat{P}_i\hat{P}_j(1 - \hat{P}_i)(1 - \hat{P}_j)}} \\ &= \frac{1}{N}\frac{1}{N-1}\left(\sum_{i=1}^{N}\sum_{j\neq i}^{N}\frac{\hat{P}_{ij}}{\sqrt{\hat{P}_i\hat{P}_j(1 - \hat{P}_i)(1 - \hat{P}_j)}} - \sum_{i=1}^{N}\sum_{j\neq i}^{N}\frac{\hat{P}_i\hat{P}_j}{\sqrt{\hat{P}_i\hat{P}_j(1 - \hat{P}_i)(1 - \hat{P}_j)}}\right)\end{aligned} \tag{6.1}$$

If we assume that $\hat{P}_i = \hat{P}_j = p$ then this simplifies to:

$$\hat{r} = \frac{1}{N^2 - N}\frac{1}{p(1-p)}\sum_{i=1}^{N}\sum_{j\neq i}^{N}\left(\hat{P}_{ij} - p^2\right) = \frac{\overline{P_{11}}/p}{1-p} - \frac{p}{1-p}$$

where $\overline{P_{11}} = \frac{1}{N^2 - N}\sum_{i=1}^{N}\sum_{j\neq i}^{N}\hat{P}_{ij}$. Writing $\overline{P_{11}}$ in terms of $\hat{r}$ and $\bar{p}$, we have:

$$\begin{aligned}\overline{P_{11}} &= \bar{p}(r(1 - \bar{p}) + \bar{p}) \\ &= \bar{p}(\hat{r} - \bar{p}r + \bar{p}) \\ &= \bar{p}((1 - r)\bar{p} + \hat{r}) \\ &= \bar{p}\frac{\bar{p} + \frac{\hat{r}}{1-\hat{r}}}{\frac{1}{1-\hat{r}}} \\ &= \bar{p}\frac{\bar{p} + \frac{\hat{r}}{1-\hat{r}}}{1 + \frac{\hat{r}}{1-\hat{r}}}\end{aligned}$$

Letting $\psi = \frac{\hat{r}}{1-\hat{r}}$ completes the definition of $\overline{P_{11}}$, the probability that a subsequent observation will match the preceding observation, for the Polya-Eggenberger model as defined by Feller (2008).

On the other hand, if $\forall i, j \in [1, N] \quad \hat{P}_i \simeq \hat{P}_j$, then we can approximate equation 6.1 using the geometric mean of the $\hat{p}$.

$$\hat{r} = \left( \frac{\prod_{i=1}^{N} \prod_{j \neq i}^{N} \left( \frac{\hat{P}_{ij}}{\sqrt{\hat{P}_i \hat{P}_j}} \right)^{\frac{1}{N^2-N}}}{\prod_{i=1}^{N} \prod_{j \neq i}^{N} \sqrt{(1-\hat{P}_i)(1-\hat{P}_j)}^{\frac{1}{N^2-N}}} - \frac{\prod_{i=1}^{N} \prod_{j \neq i}^{N} \sqrt{(\hat{P}_i \hat{P}_j)}^{\frac{1}{N^2-N}}}{\prod_{i=1}^{N} \prod_{j \neq i}^{N} \sqrt{(1-\hat{P}_i)(1-\hat{P}_j)}^{\frac{1}{N^2-N}}} \right)$$

This approximation will tend to discount extreme values where $\hat{P}_i$ is far from the arithmetic mean. Therefore, it is arguably a reasonable approximation to use since if $\hat{P}_i$ is much poorer than the other classifiers in practice we would prune classifier $i$ from the ensemble and on the other hand having $\hat{P}_i$ much greater than the accuracy of a typical ensemble member is unrealistic in practice. We can then bound $\hat{r}$ below by

$$\begin{aligned} \hat{r} &\geq \left( \frac{\prod_{i=1}^{N} \prod_{j \neq i}^{N} \left( \frac{\hat{P}_{ij}}{\sqrt{\hat{P}_i \hat{P}_j}} \right)^{\frac{1}{N^2-N}}}{\prod_{i=1}^{N} \prod_{j \neq i}^{N} \sqrt{(1-\hat{P}_i)(1-\hat{P}_j)}^{\frac{1}{N^2-N}}} - \frac{\prod_{i=1}^{N} \prod_{j \neq i}^{N} \mathrm{E}[\sqrt{(\hat{P}_i \hat{P}_j)}]^{\frac{1}{N^2-N}}}{\prod_{i=1}^{N} \prod_{j \neq i}^{N} \sqrt{(1-\hat{P}_i)(1-\hat{P}_j)}^{\frac{1}{N^2-N}}} \right) \\ &\geq \left( \frac{\prod_{i=1}^{N} \prod_{j \neq i}^{N} \left( \frac{\hat{P}_{ij}}{\sqrt{\hat{P}_i \hat{P}_j}} \right)^{\frac{1}{N^2-N}}}{\prod_{i=1}^{N} \prod_{j \neq i}^{N} \sqrt{(1-\hat{P}_i)(1-\hat{P}_j)}^{\frac{1}{N^2-N}}} - \frac{\prod_{i=1}^{N} \prod_{j \neq i}^{N} \mathrm{E}[\frac{\hat{P}_i + \hat{P}_j}{2}]^{\frac{1}{N^2-N}}}{\prod_{i=1}^{N} \prod_{j \neq i}^{N} \sqrt{(1-\hat{P}_i)(1-\hat{P}_j)}^{\frac{1}{N^2-N}}} \right) \end{aligned}$$

and above by

$$\hat{r} \leq \left( \frac{\prod_{i=1}^{N} \prod_{j \neq i}^{N} \left( \mathrm{E}[\frac{\hat{P}_{ij}}{\sqrt{\hat{P}_i \hat{P}_j}}] \right)^{\frac{1}{N^2-N}}}{\prod_{i=1}^{N} \prod_{j \neq i}^{N} \sqrt{(1-\hat{P}_i)(1-\hat{P}_j)}^{\frac{1}{N^2-N}}} - \frac{\prod_{i=1}^{N} \prod_{j \neq i}^{N} \sqrt{(\hat{P}_i \hat{P}_j)}^{\frac{1}{N^2-N}}}{\prod_{i=1}^{N} \prod_{j \neq i}^{N} \sqrt{(1-\hat{P}_i)(1-\hat{P}_j)}^{\frac{1}{N^2-N}}} \right)$$

## 6.3 Discussion on the diversity measure

The previous section shows that we can reasonably use the Polya-Eggenberger distribution to model the accuracy of a majority vote ensemble classifier when the classifiers have similar accuracy performance. Moreover, our results also show that the model is related to Sneath and Sokal's correlation measure $\rho$.

One of the weakness of using the Polya-Eggenberger distribution to model the majority vote ensemble classifier is in the assumption that the individual classifiers in the ensemble have identical accuracies. However, as our empirical results will indicate, this assumption is not crucial to good performance of the model, and that the Polya-Eggenberger model gives a very good estimate of the average accuracy of the majority vote ensemble across a very wide range of ensemble member sizes.

We define $D_i(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n) \mapsto \mathbb{R}^n$ as the indicator function for classifier $i$, with the vector $\boldsymbol{x}_m \in \mathbb{R}^d$, $m \in [1, n]$ representing the data for the $m$-th test data. We let $D_i(\boldsymbol{x}_m) = 1 - p_i$ when the classifier classifies sample point $m$ correctly and $-p_i$ otherwise with $p_i$ the expected accuracy of classifier $i$. Observe that $\rho_{i,j}$ can also be written as

$$\begin{aligned}\rho_{i,j} &= \frac{P_{ij} - p_i p_j}{\sqrt{p_i(1-p_i)p_j(1-p_j)}} \\ &= \frac{D_i(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n) \cdot D_j(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)}{\|D_i(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)\|_2 \|D_j(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)\|_2} \\ &= \cos\theta_{i,j}\end{aligned}$$

where $\theta_{i,j}$ is the angle between $D_i(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n) D_j(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$ Geometrically, this can be interpreted as the dot product between indicator functions of classifiers $i$ and $j$. See Figure 6.1 for a visual representation of this intuition. Note that this is different from the correlation measured used in Ladha (1995), in that $\rho$ is not the correlation of classifiers outputs, but the correlation of the accuracy of the classifiers. When the accuracy of the classifier is not independent of the class labels, the correlation measure used in Ladha (1995) would give us a different value from the Sneath and Sokal (1963) correlation measure. We will see later that Ladha's measure does not seem to capture the diversity as well as Sneath and Sokal.

Intuitively, in order to minimize $\rho_{i,j}$ (i.e. to increase the diversity), we should increase the number of points the classifiers disagree on — while still maintaining the overall accuracy of the classifiers. This intuition gives a plausible explanation as to the efficacy of Random Forests (Breiman, 2001). Random Forest can be seen as the combination of the Random Subspace Method with bootstrap sampling. By training the classifiers on a subset of the data, the individual classifiers of the Random Forest method would be accurate on that region of the data, and therefore would be weakly or uncorrelated to the other classifiers in the ensemble thereby giving a smaller average value for $\rho$.

It is generally accepted that the majority vote ensemble accuracy increases with increasing individual classifier accuracy, increasing ensemble member size, and decreasing correlation. One useful implications of our model is that it gives us a way to compare two classification ensembles with identical individual classifier performance but with different correlation measure and ensemble member size. Consider two

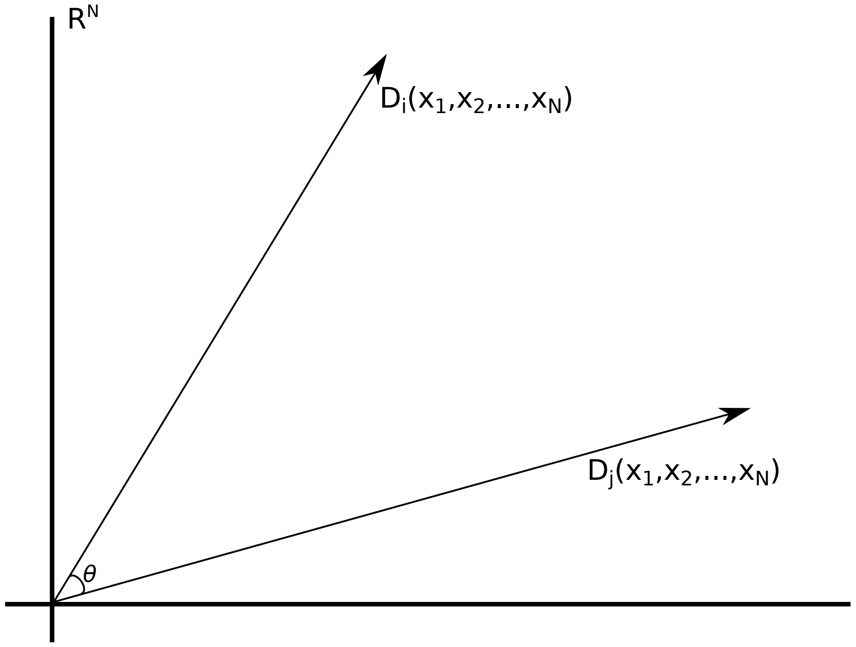

**Figure 6.1:** *Geometric interpretation of $\rho_{i,j}$. Note that $\rho_{i,j}$ can be interpreted as the cosine of the angular separation of the indicator function $D(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ that classifier $i$ classifies training example $\boldsymbol{x}_m$ correctly.*

classification ensembles with identical individual classifier performance $p > 0.5$, the first with classifier correlation $r$ and ensemble member size $N$, and the second with classifier correlation $r'$ and ensemble member size $N'$.

Using simple algebra, one can show that the variance of $\frac{S_N}{N}$ and $\frac{S'_N}{N'}$ is as given in equation 6.2 below. Now, the ensemble will typically have the better majority vote accuracy if the variance of the errors is smaller than some competing ensemble

$$\begin{aligned} \mathrm{Var}(\frac{S_N}{N}) &= p(1-p)r + (\frac{1-r}{N}) \\ \mathrm{Var}(\frac{S_{N'}}{N'}) &= p(1-p)r' + (\frac{1-r'}{N'}) \end{aligned} \tag{6.2}$$

One practical question of interest to ask is, if it is better to have fewer negatively correlated classifiers (generated via careful selection), or (infinitely) many correlated classifiers. Here, simple algebra shows that if we can generate $N > \frac{1-r}{r'-r}$, then fewer negatively correlated classifiers are better than the ensemble with $N' = \infty$ classifiers. This model can therefore be applied to help with ensemble pruning decisions such as would the accuracy of the ensemble be improved overall if we add a classifier that would lower the average accuracy but improves the diversity of the ensemble?

In the following sections, we will explore this further by using the properties and some results from concentration of measures applied to the Polya-Eggenberger distribution.

### 6.3.1 Majority Vote Accuracy as Ensemble member size $N \to \infty$

Under the assumption of our model, the Polya-Eggenberger model recovers Condorcet's Jury Theorem when average correlation of the classifiers $r$ is 0 and the average accuracy $\overline{p}$ is greater than 0.5. To see why this is so, observe that the Polya-Eggenberger distribution has the same form as a binomial distribution when $r = 0$, and therefore the accuracy of the majority vote ensemble classifier will tend to certainty as the size of the ensemble $N$ tends to infinity.

The model also implies that for all $i, j$ if $r < 0$ and $\overline{p} > 0.5$, an ensemble of size $N = \frac{1}{r} - 1$ will produce an ensemble classifier that has majority vote accuracy almost surely. This of course assumes that it is possible to produce $N = \frac{1}{r} - 1$ classifiers that are have an average correlation $r < 0$ which may not be the case in practice. The implications of the model for $r \leq 0$ are consistent with the findings of Kuncheva et al. (2000) who showed that an ensemble classifier with independent or negatively correlated classifiers will have an ensemble accuracy tending to 1 as the size of the ensemble increase.

Finally, if $r > 0$, observe that the distribution of the number of classifiers classifying correctly $S_N$ in the ensemble follows a beta-binomial distribution with $\alpha = \overline{p}\frac{1-r}{r}$ and $\beta = (1-\overline{p})\frac{1-r}{r}$ and the ensemble size $N$. Also observe that the limiting distribution for $\lim_{N\to\infty} \frac{S_N}{N}$ is the beta distribution with the shape parameters $\alpha$ and $\beta$ respectively. Here, we can use the CDF for the beta distribution given in equation 6.3 to find the asymptotic behaviour of the ensemble as the number of ensemble members $N$ goes to infinity. Figures 6.2 and 6.3 illustrates the CDF for various values of $\alpha$ and $\beta$.

We note that when $\overline{p}$ is close to 0.5, the size of the ensemble $N$ required to approach the asymptotic behaviour can be very large. The general rule of thumb is that the size of the ensemble $N$ should be at least $O(\frac{1}{(p-0.5)^2})$ before the majority vote ensemble classifier accuracy tends to the estimated asymptotic accuracy given in equation 6.3 This guideline also gives us practical considerations for capacity limited

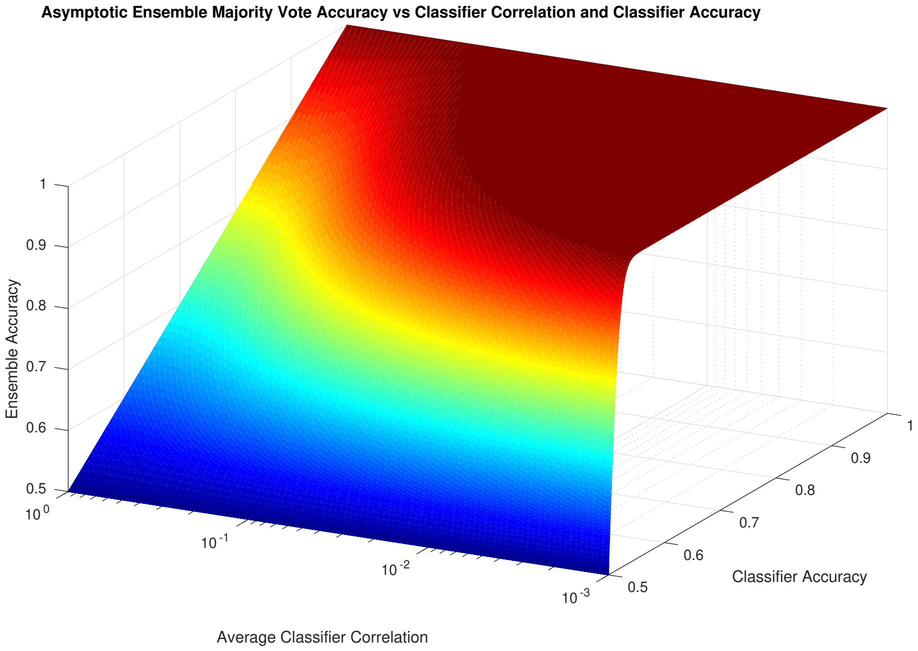

**Figure 6.2:** *Surface plot for the asymptotic accuracy of a majority vote ensemble with $N \to \infty$*

ensemble classification implementations that may have limitations in the number of classifiers.

$$\lim_{n\to\infty} \Pr\left\{\frac{S_N}{N} > 0.5\right\} = (1 - \frac{\beta_{0.5}\left(\overline{p}\frac{1-r}{r}, (1-\overline{p})\frac{1-r}{r}\right)}{\beta\left(\overline{p}\frac{1-r}{r}, (1-\overline{p})\frac{1-r}{r}\right)} \tag{6.3}$$

Of course, an ensemble classifier with infinitely many classifiers is not practically realizable. Therefore, it is also important to consider the ensemble for small values of $N$. Unfortunately, to the best of our knowledge, there is no closed form for the generalized hyper-geometric function used in the CDF of a Beta-Binomial or Polya-Eggenberger distribution, making the optimization trade-off difficult to be determined analytically. However, using results from concentration of measure, we can approximate the CDF of the Polya-Eggenberger distribution.

### 6.3.2 Analysis of the Ensemble Errors

**Proposition 6.1.** *Let $P_{i,l}$ be an indicator for classifier $i$ classify training example $l$ correctly, that is to say, $P_{i,l} = 1$ if $h_i(X^{(l)}) = y^{(l)}$ and $0$ otherwise, and let $\overline{\widehat{p}} = \frac{1}{Nn}\sum_{i=1}^{N}\sum_{l=1}^{n} P_{i,l}$ be the average empirical training accuracy. Let $p = E_{\mathcal{H}}[\overline{\widehat{p}}]$, then with probability $1-\delta$, $|p - \overline{\widehat{p}}| < \sqrt{\frac{N\tau_{\mathcal{H}}(n)+\log 2/\delta}{n}}$ where $\tau_{\mathcal{H}}(n) := \max_{C\subset X:|C|=n} |\mathcal{H}_C|$ is the growth function of hypothesis class $\mathcal{H}$ by restricting $\mathcal{H}$ to $C$.*

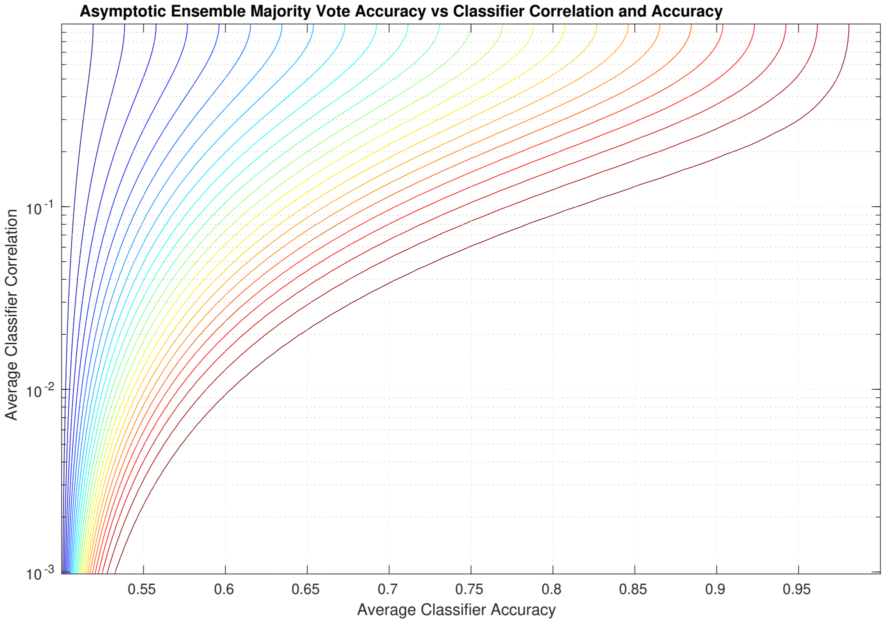

**Figure 6.3:** *Contour plot for the asymptotic accuracy of a majority vote ensemble with $N \to \infty$*

*Proof.* Let $P_{i,l}$ be an indicator for classifier $i$ classify training example $l$ correctly, that is to say, $P_{i,l} = 1$ if $h_i(X^{(l)}) = y^{(l)}$ and 0 otherwise.

Let $\widehat{p}_i = \frac{1}{n}\sum_{l=1}^{n} P_{i,l}$ be the empirical training accuracy of classifier $i$.

Let $\overline{\widehat{p}} = \frac{1}{Nn}\sum_{i=1}^{N}\sum_{l=1}^{n} P_{i,l} = \frac{1}{Nn}\sum_{l=1}^{n}\sum_{i=1}^{N} P_{i,l}$ be the average empirical training accuracy.

Applying Hoeffding's inequality and observing that $b = \max[\sum_{i=1}^{N} P_{i,l}] = N$ and $a = \min[\sum_{i=1}^{N} P_{i,l}] = 0$, we have

$$\Pr\left\{\left|p - \frac{1}{Nn}\sum_{l=1}^{n}\sum_{i=1}^{N} P_{i,l}\right| > \epsilon\right\} \leq \delta(N) = 2\exp\left(\frac{-2n^2N^2\epsilon^2}{\sum_{l=1}^{n}(b-a)^2}\right)$$
$$= \exp\left(-2n\epsilon^2\right)$$

Union bounding the inequality to $\tau_{\mathcal{H}}(n)^N$ classifiers in the hypothesis class $\mathcal{H}$ gives us

$$\Pr\left\{|p - \frac{1}{Nn}\sum_{i=1}^{N}\sum_{l=1}^{n} P_{i,l}| > \epsilon\right\} \leq 2\tau_{\mathcal{H}}(n)^N \exp\left(-2n\epsilon^2\right)$$

Solving for $\epsilon$ gives us $\epsilon = \sqrt{\frac{N\tau_{\mathcal{H}}(n)+\log 2/\delta}{n}}$,

□

Implying that with probability $1-\delta, p \geq \overline{\widehat{p}} - \sqrt{\frac{N\tau_{\mathcal{H}}(n)+\log 2/\delta}{n}}$

Assuming that $n > k+1$, then Shaur-Shelah's lemma upper bounds the size of the hypothesis class for random subspace classifiers, $\tau_{\mathcal{H}}(n) \leq (k+1)\log(n)$.

This implies that $\epsilon \leq \sqrt{\frac{N(k+1)\log(n)+\log 2/\delta}{n}} = \sqrt{\frac{(k+1)\log(n)}{n} + \frac{\log 2/\delta}{n}}$ and with probability $1-\delta, p \geq \overline{\widehat{p}} - \sqrt{N\frac{(k+1)\log(n)}{n} + \frac{\log 2/\delta}{n}}$.

Next, we use Cantelli's Inequality to lower-bound the accuracy of the majority vote ensemble.

**Proposition 6.2.** *Suppose that $p_i = p_j = p > 0.5, \forall i,j \in [1,N]$, and the number of classifiers $S_N$ classifying an arbitrary data point correctly follows a Polya-Eggenberger distribution $PE(N, p, \frac{r}{1-r})$. The probability that a majority vote ensemble classifies the point correctly is at least $Pr\left\{S_N \geq \frac{1}{2}N + cN\right\} \geq 1 - \frac{p(1-p)(r+\frac{1-r}{N})}{(p-(0.5+c))^2}$, where $c = \frac{1}{N}$ when $N$ is even, and $c = \frac{1}{2N}$ when $N$ is odd.*

*Proof.* Observe that

$$\begin{aligned}
\Pr\left\{S_N < \frac{1}{2}N + cN\right\} &= \Pr\left\{\frac{-S_N}{N} > -0.5 - c\right\} \\
&= \Pr\left\{\frac{-S_N - \mathrm{E}[-S_N]}{N} > -\frac{\mathrm{E}[-S_N]}{N} - 0.5 - c\right\} \\
&= \Pr\left\{\frac{\mathrm{E}[S_N] - S_N}{N} > \frac{\mathrm{E}[S_N]}{N} - 0.5 - c\right\}
\end{aligned}$$

Note that $\frac{\mathrm{E}[S_N]}{N} = p$ and also that

$$\mathrm{Var}[\frac{S_N}{N}] = \frac{p(1-p)(\frac{1}{r}-1)^2(\frac{1}{r}-1+N)}{N(\frac{1}{r}-1)^2((\frac{1}{r}-1+1))} = p(1-p)\frac{r}{N}(\frac{1}{r}-1+N) = p(1-p)(r+\frac{1-r}{N})$$

Then using Cantelli's inequality (Lemma 3.3), we can upper bound the misclassification rate by

$$\begin{aligned}
\Pr\left\{S_N < \frac{1}{2}N + cN\right\} &= \Pr\left\{\frac{\mathrm{E}[S_N] - S_N}{N} > \frac{\mathrm{E}[S_N]}{N} - 0.5 - c\right\} \\
&\leq \frac{\mathrm{Var}[\frac{S_N}{N}]}{\mathrm{Var}[\frac{S_N}{N}] + (\frac{\mathrm{E}[S_N]}{N} - 0.5 - c)^2} \\
&= \frac{p(1-p)(r+\frac{1-r}{N})}{p(1-p)(r+\frac{1-r}{N}) + (p-(0.5+c))^2}
\end{aligned}$$

$$\begin{aligned}
\Pr\left\{S_N \geq \frac{1}{2}N + cN\right\} &\geq 1 - \frac{p(1-p)(r+\frac{1-r}{N})}{p(1-p)(r+\frac{1-r}{N}) + (p-(0.5+c))^2} \\
&= \frac{(p-(0.5+c))^2}{p(1-p)(r+\frac{1-r}{N}) + (p-(0.5+c))^2} \\
&\geq 1 - \frac{p(1-p)(r+\frac{1-r}{N})}{(p-(0.5+c))^2}
\end{aligned}$$

□

With the last inequality coming from upper bounding the inequality using the first two terms of the geometric series expansion. Combining proposition 6.1 and 6.2, gives us an estimate of the ensemble accuracy based on the empirical estimation of the classifiers by substituting $p$ with $\bar{\hat{p}} - \sqrt{\frac{N\tau_{\mathcal{H}}(n)}{n} + \frac{\log(2/\delta)}{n}}$.

### 6.3.3 'Good' and 'Bad' diversity error decomposition

The results of proposition 6.2 can be extended to the derive the 'Good' and 'Bad' diversity error decomposition from Brown and Kuncheva (2010). Recall from section 2.1.5, that classifier error can be decomposed into

$$\mathrm{E}\left[L(f_{ens} - y)\right] = \int_{\boldsymbol{x}} L(f_i - y) + \underbrace{\int_{\boldsymbol{x}^-} \frac{1}{N}\sum_i^N L(f_i - f_{ens})}_{\text{“Bad Diversity”}} - \underbrace{\int_{\boldsymbol{x}^+} \frac{1}{N}\sum_i^N L(f_i - f_{ens})}_{\text{“Good Diversity”}} \tag{6.4}$$

Now we can rewrite equation 6.4 in terms of $p_i$ as

$$\begin{aligned}\mathrm{E}\left[L(f_{ens} - y)\right] = &\sum_{i=1}^N (1 - p_i) + \underbrace{\frac{1}{N}\sum_{i=1}^N \Pr\left\{\frac{S_N}{N} < 0.5 + c|P_1 = 1\right\} p_i}_{\text{“Bad Diversity”}} \\ &\underbrace{- \frac{1}{N}\sum_i^N \Pr\left\{\frac{S_N}{N} > 0.5 + c|P_1 = 0\right\}(1 - p_i)}_{\text{“Good Diversity”}}\end{aligned} \tag{6.5}$$

where $c = \frac{1}{N}$ when $N$ is even, and $c = \frac{1}{2N}$ when $N$ is odd. Under our Polya-Eggenberger model, we have

$$\left[\frac{S_N}{N}|P_1 = 1\right] \sim \frac{\mathrm{PE}(N-1, p + (1-r)p, \frac{r}{1-r}) + 1}{N}$$

and,

$$\left[\frac{S_N}{N}|P_1 = 0\right] \sim \frac{\mathrm{PE}(N-1, p - rp, \frac{r}{1-r})}{N}$$

Therefore,

$$Var\left[\frac{S_N}{N}|P_i = 1\right] = (p + r(1-p))(1 - p - r(1-p))(r + \frac{1-r}{N-1}) \leq \frac{1}{4}(r + \frac{1-r}{N-1})$$

and similarly,

$$Var\left[\frac{S_N}{N}|P_i = 0\right] = (p - rp))(1 - p + rp))(r + \frac{1-r}{N-1}) \leq \frac{1}{4}(r + \frac{1-r}{N-1})$$

Using Cantelli's inequality, an upper bound for the "bad diversity" is

$$\begin{aligned}
&\frac{1}{N}\sum_{i=1}^{N}\Pr\left\{S_N < \frac{1}{2}N + cN|P_i = 1\right\}p_i\\
&\quad= \frac{1}{N}\sum_{i=1}^{N}\Pr\left\{\frac{\mathrm{E}[S_N]-S_N}{N} > \frac{\mathrm{E}[S_N]}{N} - 0.5 - c\right\}p_i\\
&\quad\le \frac{1}{N}\sum_{i=1}^{N}\frac{p_i\mathrm{Var}[\frac{S_N}{N}|P_i=1]}{\mathrm{Var}[\frac{S_N}{N}|P_i=1] + (p + r(1-p) + \frac{1}{N} - 0.5 - c)^2}\\
&\quad\le \frac{1}{N}\sum_{i=1}^{N}\frac{p_i(r + \frac{1-r}{N-1})}{(r + \frac{1-r}{N-1}) + 4(p + r(1-p) + \frac{1}{N} - 0.5 - c)^2}\\
&\quad\le \frac{1}{N}\sum_{i=1}^{N}\frac{p_i(r + \frac{1-r}{N-1})}{4(p + r(1-p) + \frac{1}{N} - 0.5 - c)^2}\\
&\quad= \frac{p(r + \frac{1-r}{N-1})}{4(p + r(1-p) + \frac{1}{N} - 0.5 - c)^2}
\end{aligned} \tag{6.6}$$

Similarly, a lower bound for the "good" diversity is

$$\begin{aligned}
&\frac{1}{N}\sum_{i=1}^{N}\Pr\left\{S_N > \frac{1}{2}N + cN|P_i = 0\right\}(1-p_i)\\
&\quad= (1-p)(1 - \Pr\left\{\frac{\mathrm{E}[S_N]-S_N}{N} > \frac{\mathrm{E}[S_N]}{N} - 0.5 - c\right\}\\
&\quad\ge (1-p)\left(1 - \frac{\mathrm{Var}[\frac{S_N}{N}|P_i=1]}{\mathrm{Var}[\frac{S_N}{N}|P_i=1] + (p - rp - 0.5 - c)^2}\right)\\
&\quad\ge (1-p)\left(1 - \frac{r + \frac{1-r}{N-1}}{(r + \frac{1-r}{N-1}) + 4((p - rp) - 0.5 - c)^2}\right)\\
&\quad= (1-p)\left(\frac{4((p - rp) - 0.5 - c)^2}{(r + \frac{1-r}{N-1}) + 4((p - rp) - 0.5 - c)^2}\right)
\end{aligned} \tag{6.7}$$

Taking the first 2 terms of the geometric series expansion gives us

$$\frac{4(1-p)(p - rp - 0.5 - c)^2}{r + \frac{1-r}{N-1}}\left(1 - \frac{(r + \frac{1-r}{N-1})}{4(p - rp - 0.5 - c)^2}\right)$$

Replacing equation 6.6 and equation 6.7 into equation 6.5 gives us

$$\begin{aligned}
\mathrm{E}\left[L(f_{ens} - y)\right] \le& 1 - p + \underbrace{\frac{p(r + \frac{1-r}{N-1})}{4(p + r(1-p) + \frac{1}{N} - 0.5 - c)^2}}_{\text{“Bad” Diversity}}\\
&- \underbrace{\frac{4(1-p)(p - rp - 0.5 - c)^2}{r + \frac{1-r}{N-1}}\left(1 - \frac{(r + \frac{1-r}{N-1})}{4(p - rp - 0.5 - c)^2}\right)}_{\text{“Good” Diversity}}
\end{aligned} \tag{6.8}$$

The implication of this decomposition is consistent with our intuitions and the generally accepted results stated in the previous sections. The ensemble error is minimized as the classifier correlation $r$ decreases, ensemble member size $N$ increases and average individual classifier accuracy $p$ approaches 1. We note however that the

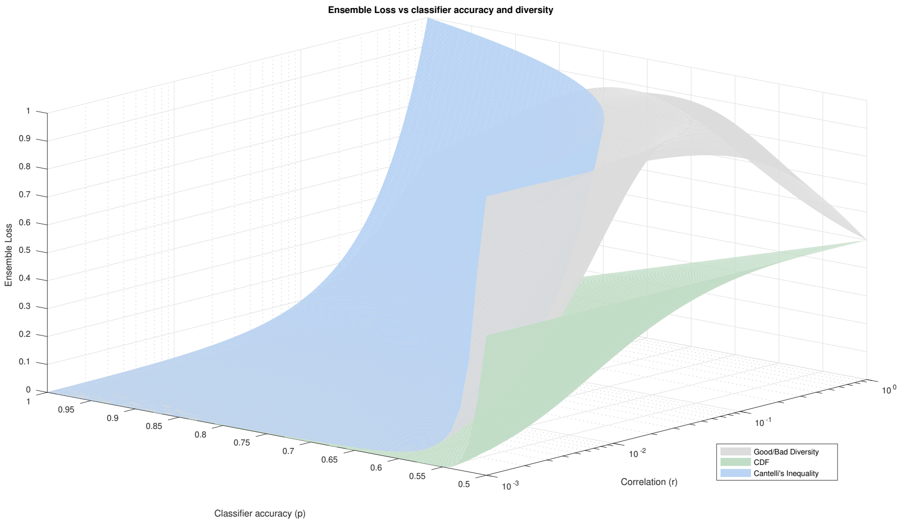

**Figure 6.4:** *Surface plot of the ensemble 0-1 loss versus average member classifier accuracy and classifier correlation according to the CDF, "Good/Bad" diversity error decomposition (equation 6.8) and Cantelli's Inequality. Observe that the CDF is bounded above by the Cantelli's Inequality and Error Decomposition. Note that the non-monotonic behaviour of the "Good/Bad" error decomposition comes from using Cantelli's Inequality to approximate the CDF.*

convex combination of the decomposition implied that the error function increases for certain values of $r$. This is however an artefact of using Cantelli's inequality, and the behaviour of the error function is monotonic to both $r$ and $p$. This misleading behaviour goes away when the CDF of Polya-Eggenberger distribution is used instead of approximating using Cantelli's inequality. However, because we were unable to make headway into giving a closed form to the generalized hyper-geometric function, this remains for future research.

## 6.4 Soft-voting / Sum rule

An alternative approach to majority voting is soft-voting, or sometimes called sum or average rule. Under soft voting, rather than each classifier voting for one class over another, the classifier outputs a score reflecting the"confidence" that an observation belongs to a given class. The additional information gained by knowing how confident the classifier is on the class label can sometimes lead to improved accuracy. Let $\widehat{h}_i$ be the normal vector to the hyperplane representing linear classifier ensemble member $i$ and let $h$ be the Bayes' classifier in the hypothesis class $\mathcal{H}$. Since $\widehat{h}_i$ is learnt using a random subspace projection that is independent of the need for $\hat{h}_j$, it follows that $\widehat{h}_i$ is independent of $\hat{h}_j$ as well, moreover, $\mathrm{E}[\hat{h}_i] = \mathrm{E}[\hat{h}_j], \forall i, j$. The decision rule of the ensemble classifier using soft vote would then be

$$\mathbf{1}\left(\sum_{i=1}^{n} \hat{h}_i^T x > 0\right) = \mathbf{1}\left((\sum_{i=1}^{n} \hat{h}_i)^T x > 0\right)$$

The probability of misclassification can then be written as

$$\Pr\left\{\frac{\sum_{i=1}^{N} \hat{h}_i^T x}{h^T x} \leq 0\right\} = \Pr\left\{\left|\frac{1}{n}\sum_{i=1}^{N} \hat{h}_i^T x - \mathrm{E}[\hat{h}_1^T]x\right| > h^T x]\right\}$$

If we assume $\lim\limits_{N\to\infty} P[|\frac{1}{N}\sum_{i=1}^{N} h_i^T x - h^T x]| > \epsilon] = 0$, (i.e. the classifiers weakly converge to the Bayes classifier, for example $\hat{h}_i = \boldsymbol{P}_i h$ where $\boldsymbol{P}_i$ is the $i$-th RS projection then this is clearly true) then by Hoeffding's inequality we have

$$\Pr\left\{|\frac{1}{N}\sum_{i=1}^{N}(\hat{h}_i^T x - \mathrm{E}[h_i]^T x)| > h^T x\right\} \leq 2\exp\left(\frac{-N^2(h^T x)^2}{\sum_{i=1}^{N} c_i^2}\right)$$

where $c_i = |h_i^T x - h^T x| \leq C$. Without loss of generality, if we also assume that $\|h_i\| = 1$, and $\|x\| = 1$, otherwise we normalize by dividing $h_i$ by $\|h_i\|$ and $x$ by $\|x\|$, then $C \leq 2$ by Cauchy-Schwatz inequality and we have

$$\Pr\left\{\left|\frac{1}{N}\sum_{i=1}^{N}(\hat{h}_i^T x - \mathrm{E}[h_i]^T x)\right| > h^T x\right\} \leq 2\exp\left(\frac{-N(h^T x)^2}{4}\right)$$

Finally, the error of the sum-rule classifier can be written as

$$\mathrm{E}[\mathbf{1}\{\frac{1}{N}\sum_{i=1}^{N}(\hat{h}_i^T x)\} \neq y] \leq 2\exp\left(\frac{-N(h^T x)^2}{4}\right) + \mathrm{E}[\mathbf{1}\{h^T x\} \neq y]$$

This shows the accuracy of a sum-rule classifier improves as the size of the ensemble increases. Moreover, our results implies also that as the ensemble size $N$ tends to $\infty$, we recover the Bayes' classifier as suggested by our result from section 5.4.

### 6.4.1 Other estimates of the diversity measures in the Polya Distribution

In much of the ensemble classification literature (e.g. Malmasi and Dras (2015); Whalen and Pandey (2013); Yang (2011)), the Yule's Q-statistic (Yule, 1900) is used to measure the diversity of the classifiers in the ensemble. It is not surprising why many researchers favour using the Q-statistics, as the measure has an intuitive interpretation as the odds-ratio and is specifically designed for discrete counts (Kuncheva et al., 2000). We found however, that this diversity measure tends to overestimate the correlation and can be widely off the mark when used with our Polya-Eggenberger model.

One shortcoming of both the Sneath and Sokal (1963) diversity measure and Yule (1900) Q-statistics is that we need to know beforehand the classifier accuracy performance before we can determine the diversity. It may be helpful to be able to estimate the diversity of the classifiers independently of classifier accuracy, such as when evaluating diversity generation schemes. In both Ladha (1995) and Berg (1993), the authors used $r = corr(\boldsymbol{y}_i, \boldsymbol{y}_j)$ to estimate the diversity measure, where $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ are the vectors representing the output labels of classifier $i$ and $j$ respectively. While we agree that voting agreement is natural and intuitive to derive the increased likelihood that two voters would vote similarly, we found that this measure also tends to be overly conservative (even more so than the Q-statistic) and overestimates the correlation of the classifiers which we suspect is due to the accuracies of the classifiers not being independent of the class labels.

We also find that it may be useful to be able to estimate the correlation $\rho$ before generating the individual classifiers though the learning algorithm. As noted by Sun and Zhou (2018), a "structural" diversity measure should be considered in addition to a "behavioral" diversity for ensemble methods since "behavioral" diversity may just be another appearance of accuracy and it is difficult to encourage "behavioral" diversity explicitly. Inspired by the feature stability measures used for feature selection (Nogueira and Brown, 2015), we also evaluate the Jaccard similarity index as an estimate for $\rho$. Conceptually the Jaccard similarity index can be viewed as the proportion of shared features over the number features in the classifiers. This intuition has some basis according to the intuition behind the "wisdom" of crowds, in that, the additional decision makers should add to the collective information of the group, and the Jaccard similarity captures the "similarity" in the structure of the information contributed by each classifier. For random subspace that the expected Jaccard similarity index is

$$\mathrm{E}[J_{i,j}] = \frac{\frac{k^2}{d^2}}{2\frac{k}{d} - \frac{k^2}{d^2}} = \frac{k}{2d - k}$$

From our observation, we found that the Jaccard similarity index gives a good estimate for our synthetic cases but gives an incorrect estimate on the real-world test cases.

## 6.5 Empirical Corroboration

### 6.5.1 Synthetic Data

We set $d = 1000$ to be the dimensionality of our data, $\boldsymbol{u} = (1, 0, \ldots, 0)$ and draw $\boldsymbol{t}_i \sim (0, N(0, I_{d-1})/\|\boldsymbol{t}_i\|$. Note that $\boldsymbol{u}$ is orthogonal to $\boldsymbol{t}_i$ We then generate $\boldsymbol{R}$ as an orthonormalized rotation matrix with entries

$$R_{i,j} \sim \frac{1}{\sqrt{d}} N(0, I_d)$$

Observe that $\boldsymbol{R}$ is analogous to a random rotation matrix that rotates $\boldsymbol{u}$ and $\boldsymbol{t}_i$ along $d$ coordinates (see Figure 6.5 for a visualization of the transformation). Also observe that $\|\boldsymbol{uR}\|_4^4 \approx 3$ and $\|\boldsymbol{uR} \odot \boldsymbol{t}_i\boldsymbol{R}\|_2^2 \approx 1$. For each $\theta = \{80°, 85°, 87.5°\}$, we then let $\boldsymbol{h} = \boldsymbol{uR}$, and $\boldsymbol{x}_i = (\boldsymbol{u}\cos\theta + \boldsymbol{t}_i \sin\theta)\boldsymbol{R}$. By construction, $\boldsymbol{h}$ and $\boldsymbol{x}_i$ has an angular separation exactly $\theta$. Observe that $\boldsymbol{h}$ also describes the Bayes' optimal classifier which separates the two classes perfectly.

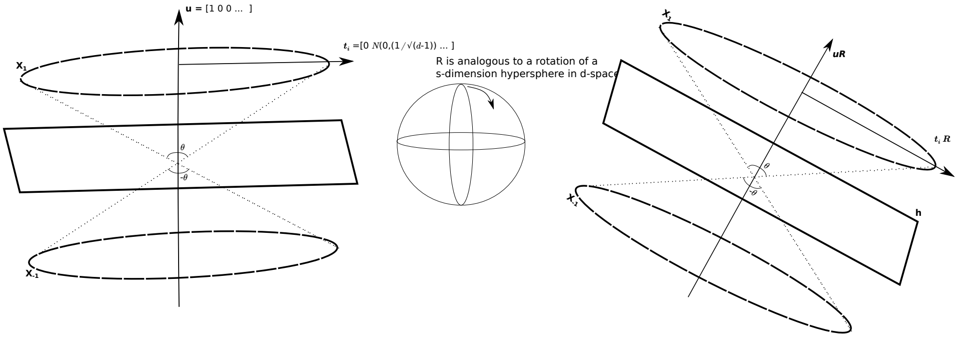
u = [1 0 0 ... ]
$t_i$ =[0 N(0,(1/√(d-1)) ... ]
R is analogous to a rotation of a s-dimension hypersphere in d-space
$X_1$
$\theta$
$-\theta$
$X_{-1}$
$X_1$
uR
$t_i\,R$
$\theta$
$-\theta$
$X_{-1}$
h

**Figure 6.5:** *Visual representation of the data after a random rotation in d-dimensions*

As noted in chapter 5, $\theta$ can be interpreted as the difficulty of the classification problem with a value of $\theta$ that is closer to $\pi/2$ representing a more difficult problem with a "smaller margin" separating the two classes.

We repeat $n_{\text{train}} \in \{150, 500, 2000\}$ draws of the training examples of $\boldsymbol{t}_i$ and set $T_n := \{\boldsymbol{x}^{(i)} \in R^d\}_{i=1}^n$ be the set of $n_{\text{train}}$ with exactly $n_{\text{train}}/2$ examples with angular separation $\theta$ and $n_{\text{train}}/2$ examples with angular separation $-\theta$. We label

$\boldsymbol{y}^{(i)} = 1$ if the corresponding $\boldsymbol{x}^{(i)}$ has an angular separation $\theta$ and $\boldsymbol{y}^{(i)} = -1$ if the angular separation is $-\theta$. Using the same data generation scheme, we also generate an additional $n_{\text{val}} = 1000$ and $n_{\text{test}} = 1000$ hold out and test examples, with an angular separation of $\theta$ and $-\theta$ and the corresponding class labels $\{1, -1\}$, divided evenly within the two datasets. We let $\boldsymbol{X}_{\text{train}}, \boldsymbol{X}_{\text{val}}$ and $\boldsymbol{X}_{\text{test}}$ be the data matrix representing the training, holdout, and test data respectively.

We then learn $N = 250$ random subspace projected classifiers on the $\boldsymbol{X}_{\text{train}}$. We first project $\{\boldsymbol{x}\}^{n_{\text{train}}}$ with $\boldsymbol{P}_j$, where $\boldsymbol{P}_j$ is a random subspace projection from $\mathbb{R}^d \mapsto \mathbb{R}^k$ and learn the classifiers $\hat{h_j}$ using linear discriminant analysis routine provided by Matlab Central (Dwinnell, 2010).

We also measured the empirical accuracy for the weighted majority vote and weighted sum rule versus the unweighted majority vote and sum rules. While using weighted majority schemes have shown good performance in some problem domains (Schapire, 1990; Blum, 1997), Kuncheva and Rodríguez (2014) showed there was no statistical difference between using a weighted versus non-weighted majority voting scheme. We therefore try to reconcile these two contradictory findings, and as an aside, evaluate the difference between the four schemes.

We set the weights $w_j = \log(P_{val}/(1 - P_{val}))$ with $P_{val}$ the accuracy of $\hat{h_j}$ on the hold-out validation set.

We measure the following empirical accuracies, where $N(A)$ is count returning the cardinality of $A$.

- $P_{majVote} := \frac{N((\sum_{j=1}^{m}(\mathbf{1}(\hat{h_j}^T P_j X_{\text{test}})))/h^T X_{\text{test}}>0)}{N(X_{\text{test}})}$
- $P_{softVote} := \frac{N((\mathbf{1}(\sum_{j=1}^{m}(\hat{h_j}^T P_j X))/h^T X_{\text{test}}>0)}{N(X_{\text{test}})}$
- $P_{weightedMajVote} := \frac{N((\sum_{j=1}^{m}(w_j \mathbf{1}(\hat{h_j}^T P_j X_{\text{test}})))/h^T X_{\text{test}}>0)}{N(X_{\text{test}})}$
- $P_{weightedSoftVote} := \frac{N((\mathbf{1}(\sum_{j=1}^{m}(w_j \hat{h_j}^T P_j X_{\text{test}}))/h^T X_{\text{test}}>0)}{N(X_{\text{test}})}$

We repeat the experiments thirty times to ensure that the results we obtain is consistent and we plot the empirical accuracies, as shown in figures 6.6 – 6.9. As reference, we also plotted the classifier learnt by the learning algorithm using the training data without RS projection (hereby denoted as the "base classifier").

### 6.5.2 RS Ensembles for noisy data

We would also like to observe is the performance of RS Ensembles in the presence of noise, in particular, feature noise (irrelevant features with no explanatory power) and label noise (mislabelled training examples).

We follow the experimental setup in subsection 6.5.1, however we now let $1/s$ be the proportion of relevant features in the data to the number of features in the data $d$, with $s = \{1, 4, 10\}$. We then redefine the rotation matrix $\boldsymbol{R}$ as below

$$\boldsymbol{R} := \left[\begin{array}{c|c} \frac{1}{\sqrt{d/s}} N(0, I_{d/s}) & 0 \\ \hline 0 & I_{d(1-\frac{1}{s})} \end{array}\right]$$

Geometrically, we can interpret $d/s$ as the number of "relevant features" with explanatory power, and a higher value for $s$ giving us a "noisy" dataset with $d - d/s$ irrelevant features. Observe also here that $\|\boldsymbol{uR}\|_4^4 \approx 3s$ and $\|\boldsymbol{uR} \odot \boldsymbol{t}_i\boldsymbol{R}\|_2^2 \approx 1$.

We then evaluate the RS ensemble's robustness to label noise. We set $q = \{0, 0.05, 0.25\}$ to be the proportion of mislabelled data in the training set. We added label noise using the settings below:

- Both classes mislabelled at random with probability $q$ on both the training data and the holdout data
- Both classes mislabelled at random with probability $q$ on the training data, holdout data is not mislabelled
- Class 1 is mislabelled at random with probability $2q$ on the training data and the holdout data. Class $-1$ is labelled perfectly.

As before in section 6.5.1, we generated $n_{\text{train}} \in \{150, 500, 2000\}$ training examples $n_{\text{val}} = \{1000\}$ and $n_{\text{test}} = 1000$ hold out and test data and measured the empirical accuracies of

- $P_{majVote}(m) := \frac{N((\sum_{j=1}^{m}(\mathbf{1}(\hat{h_j}^T P_j X_{\text{test}})))/h^T X_{\text{test}}>0)}{N(X_{\text{test}})}$
- $P_{softVote}(m) := \frac{N((\mathbf{1}(\sum_{j=1}^{m}(\hat{h_j}^T P_j X))/h^T X_{\text{test}}>0)}{N(X_{\text{test}})}$
- $P_{weightedMajVote}(m) := \frac{N((\sum_{j=1}^{m}(w_j \mathbf{1}(\hat{h_j}^T P_j X_{\text{test}})))/h^T X_{\text{test}}>0)}{N(X_{\text{test}})}$
- $P_{weightedSoftVote}(m) := \frac{N((\mathbf{1}(\sum_{j=1}^{m}(w_j \hat{h_j}^T P_j X_{\text{test}}))/h^T X_{\text{test}}>0)}{N(X_{\text{test}})}$

### 6.5.3 Empirical corroboration for the Polya-Eggenberger model

We extracted the Sneath and Sokal (1963) diversity measure $\rho$ and the average individual classifier accuracy $p$ for the Polya-Eggenberger model using the empirical results from the experiments in section 6.5.1 and 6.5.2. Tables 6.2 through 6.5 summarizes the parameters obtained from the simulation. As reference, we also extracted the $Q$-statistics, the vote correlation score, and the Jaccard similarity index to compare against the $\rho$ diversity measure.

We then calculated the CDF of the Polya-Eggenberger model and using the different estimates of the correlation, we plot the predicted ensemble accuracy modelled by the Polya-Eggenberger distribution. Next, we overlaid the empirical majority vote ensemble accuracy averaged over 30 runs obtained from the previous section to be compared against our model. For comparison, we also plotted the estimated accuracy of the ensemble as predicted by a binomial model, and the estimated asymptotic accuracy of the ensemble as $N \to \infty$ determined by our theory.

Figures 6.17 through 6.20 shows the predicted ensemble accuracy modelled by the Polya-Eggenberger Distribution with the different determination of $\psi$ against the empirical majority voting ensemble accuracy averaged over 30 runs.

Tables 6.6 shows the comparison between the average majority vote ensemble classifier accuracy for ensemble size $N = 50$, $N = 100$ and $N = 250$ against the values predicted by the Polya-Eggenberger Model using the Sneath and Sokal (1963) diversity measure as the estimate for $\psi$.

### 6.5.4 Discussion

In general, we observed that the accuracy of the classification ensembles increases with increasing ensemble size $N$ and subspaces $k$. We also observe that the accuracy decreases with increasing angles between the data and the vector $\boldsymbol{h}$, and the accuracy improves with when the number of training examples $n_{train}$ increases (see figures 6.10. For large values of $\theta$, (i.e. “difficult” problems), number of training examples needed to generalize the problem increases as predicted by Kabán and Durrant (2017). This behaviour is most clearly seen in figure 6.11.

For small values of the subspaces $k$, the ensemble appears to be affected by the feature noise $s$ with increasing $s$, reducing the overall accuracy of the ensemble.

This trend goes away with a larger number of subspaces, and is consistent with our results in chapter 5, which shows that for the 'base' flip probability for very 'sparse' vectors, reduces when $k$ is large. This is consistent with our results from chapter 4, since datasets with large feature noise $s$ would also have large regularity constants $c'$ ($c' \propto s$). As such, the classifiers would require larger number of subspaces in order to give an accurate ensemble. When we contrast figure 6.12, and 6.13 we observed that the accuracy of an ensemble of small subspace classifiers is much lower when the "sparsity" (i.e. many irrelevant features) is large, while the ensemble is able to achieve very good accuracy when the "sparsity" is low. This behaviour is not seen in ensembles with larger number of subspaces.

Figures 6.6–6.9 shows the accuracy of a weighted and unweighted ensembles under a majority vote and soft vote, where we see that the weighted majority vote and weighted sum vote ensembles classifiers appears to be significantly more accurate in comparison to the unweighted ensemble classifiers when the feature noise $s$ is large. The advantage of using a weighted scheme disappears when the feature noise is small. This can plausibly explain the contradictory findings between Kuncheva et al. (2000) and (Schapire, 1990; Blum, 1997). Interestingly, this gain from using a weighted scheme is larger than the accuracy loss from having many irrelevant features, even when the number of subspaces is small.

In general, we also observed that the accuracy of the sum rule classifier improves significantly when the number of training examples increases. This behaviour is consistent with our theory in section 6.4. This trend is most clearly seen in figure 6.14, where we can see that the sum rule classifier improving much more than the other combination schemes as the number of training examples increases.

In general, we also observed that the accuracy of the base classifier (i.e. the classifier learnt with all the features) performed extremely poorly when the number of training examples are smaller than the dimensionality of the data. This is unsurprising is known as the curse of dimensionality. However, in figures 6.6, 6.7 we see that in spite of the small training examples, the LDA algorithm was able to generate consistent member classifiers that when ensembled, generalized well to the problem. This is similar to the findings by Durrant and Kabán (2014) on RP where it was shown that RP ensembles can be used as regularization in linear classifiers. Intuitively, we can see why this would also be the case for RS ensembles. The RS projection of $\boldsymbol{X}$ is simply the sub-matrix of $\boldsymbol{X}$ with $k$ columns retained. As long as each of the $k$ columns of $\boldsymbol{X}$ is linearly independent from the other $k-1$ columns, the sub-matrix will be full rank, and the findings of Durrant and Kabán (2014). Unfortunately, to the best of our knowledge, there is no way to guarantee that each of the $k$ columns cannot be written as a linear combination of $k-1$ other columns other than to exhaustively check the matrix.

However, Tropp (2009) provides a randomized algorithm that produces an invertible matrix from a sub-sample of $k$ columns from $d$ columns of $\boldsymbol{X}$ with probability at least $3/4$ provided that the stable rank $sr(\boldsymbol{X}) = \frac{\|\boldsymbol{X}\|^2_{\mathrm{Fro}}}{\|\boldsymbol{X}\|^2}$ is larger than $ck$, and a condition number $\kappa(\boldsymbol{X}^T\boldsymbol{X}) \leq \sqrt{3}$. This implies that the stable rank can be used as a measure to determine if the random subspace projected matrix will be full ranked.

We observed very little difference on the majority vote ensemble classifier performance on the different noise settings setting. Furthermore, our empirical result also show that RS ensembles are robust to label noise. This can be seen in figure 6.15, where we see that the classifiers performed similarly in the presence of label noise. We note however that the robustness to label noise appears to be dependent on the number of training examples, and that the classifiers have better robustness when a larger number of training examples is provided, (see figure 6.16). This result lends further credence that RS ensembles can be used as a regularizer. We will leverage on these two facts later in section 7.4 where we consider adversarial examples and evaluate the ensemble's robustness against adversarial examples.

We observed that the Polya-Eggenberger model using the Sneath and Sokal (1963) diversity measure is a very good estimate for the average majority vote ensemble classifier accuracy. The difference between the majority vote accuracy is almost imperceptible as seen in figures 6.17–6.20, with the accuracy of model increasing as the number of subspaces in the classifiers increases. The largest difference happens when the ensemble member size is approximately $N/2 = 125$ and the feature noise $S$ is large (i.e. $S = 10$), for small classifier subspace sizes $k = 2$, in which case the absolute empirical majority vote accuracy differs by less than 2% compared to the

Polya-Eggenberger model. This is in spite, not having the conditions of the models (i.e. identical classifier accuracies) satisfied. Indicating to us that the assumption of identical member classifier accuracy may be relaxed. The difference between the Polya-Eggenberger model and the empirical results are not statistically significant. Tables 6.7 and 6.6 shows the numerical comparison between the model and empirical accuracies for $k = 2$.

We observed that none of the other estimator provides a consistent estimator for the correlation measure to the distributions. In general, the other estimator tends to overestimate the correlation, however it is unclear when or the conditions when estimator overestimates the correlation. Finding a diversity measure estimate that does not require knowing the individual classifier accuracy performance remains an open problem.

| | | k=2 | | | k=10 | | | k=50 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| angle | n | p | std(p) | r | p | std(p) | r | p | std(p) | r |
| 80 | 150 | 57.84 | 0.33 | 0.0005 | 68.31 | 0.26 | 0.0012 | 82.63 | 0.29 | 0.0030 |
| | 500 | 58.35 | 0.37 | 0.0004 | 69.92 | 0.26 | 0.0006 | 87.72 | 0.19 | 0.0015 |
| | 2000 | 58.54 | 0.32 | 0.0003 | 70.34 | 0.28 | 0.0004 | 89.27 | 0.13 | 0.0008 |
| 85 | 150 | 53.21 | 0.16 | 0.0008 | 57.57 | 0.18 | 0.0028 | 64.86 | 0.31 | 0.0098 |
| | 500 | 53.88 | 0.17 | 0.0006 | 59.43 | 0.21 | 0.0014 | 70.35 | 0.23 | 0.0055 |
| | 2000 | 54.16 | 0.18 | 0.0003 | 60.29 | 0.14 | 0.0005 | 72.74 | 0.18 | 0.0017 |
| 87.5 | 150 | 51.04 | 0.12 | 0.0010 | 52.48 | 0.16 | 0.0045 | 55.04 | 0.18 | 0.0167 |
| | 500 | 51.52 | 0.12 | 0.0008 | 53.72 | 0.13 | 0.0032 | 58.35 | 0.13 | 0.0142 |
| | 2000 | 51.90 | 0.13 | 0.0005 | 54.76 | 0.13 | 0.0013 | 60.96 | 0.16 | 0.0060 |

**Table 6.2:** *Parameter for Polya-Eggenberger model estimated from empirical simulation on a noiseless setting (mislabelling proportion $q = 0$ and feature noise $s = 1$). Values of $p$ is given in percentage.*

## 6.6 Application on UCI Datasets

We use five non-synthetic datasets, taken from the 2003 NIPS feature selection challenge, namely GISETTE, ARCENE, DEXTER, DOROTHEA, and MADELON (Guyon et al., 2004). Table 6.8 summarizes the characteristics of the dataset.

We removed the features that have zero variance in the training set. As in the synthetic datasets, we applied RS projection with a fixed number of subspaces ($k$)

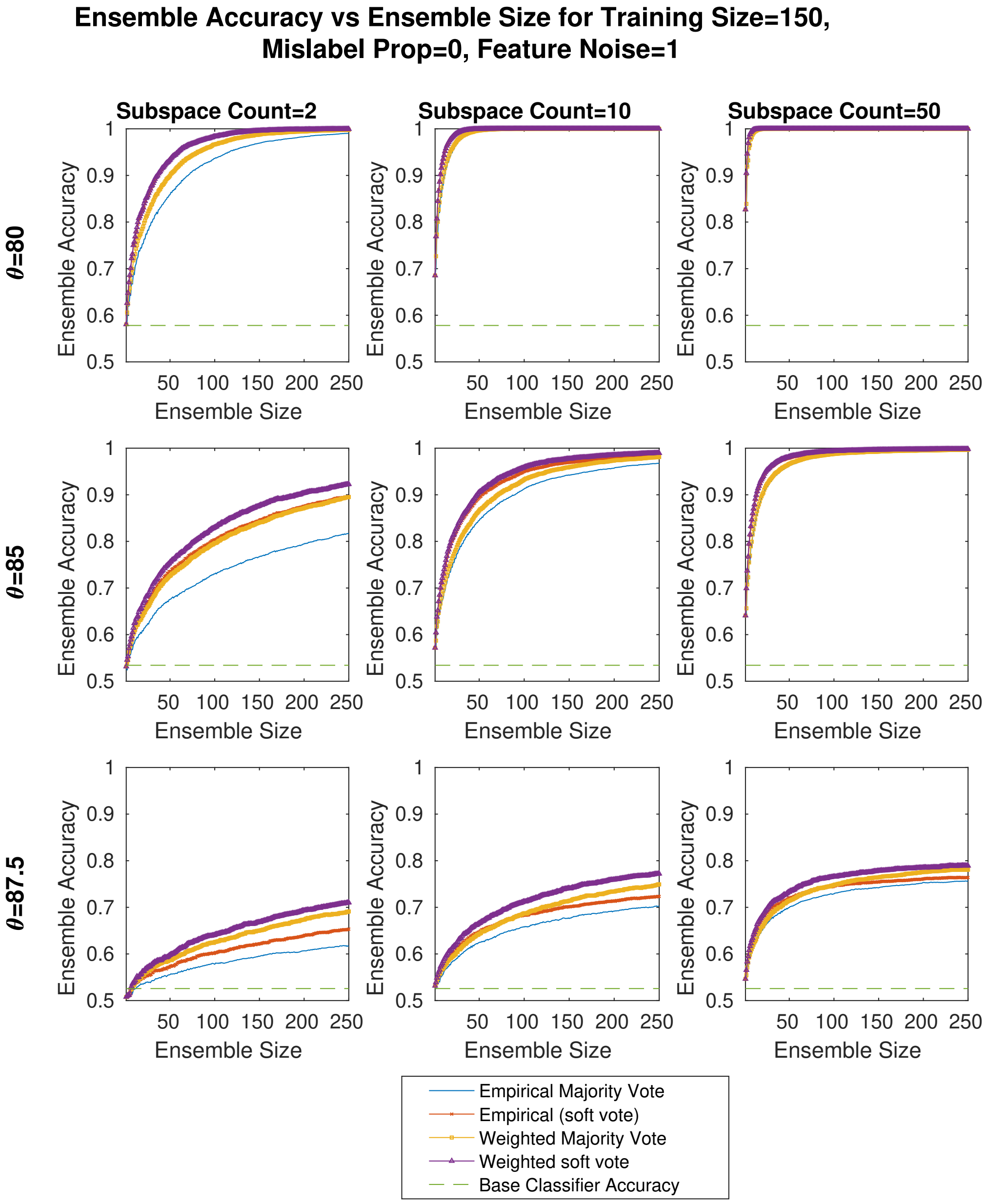

**Figure 6.6:** *Ensemble classification accuracy vs ensemble member size for a noiseless setting (feature noise proportion $s = 1$, mislabel proportion $q = 0$), with low number of training examples $n = 150$ and dimensionality $d = 1000$*

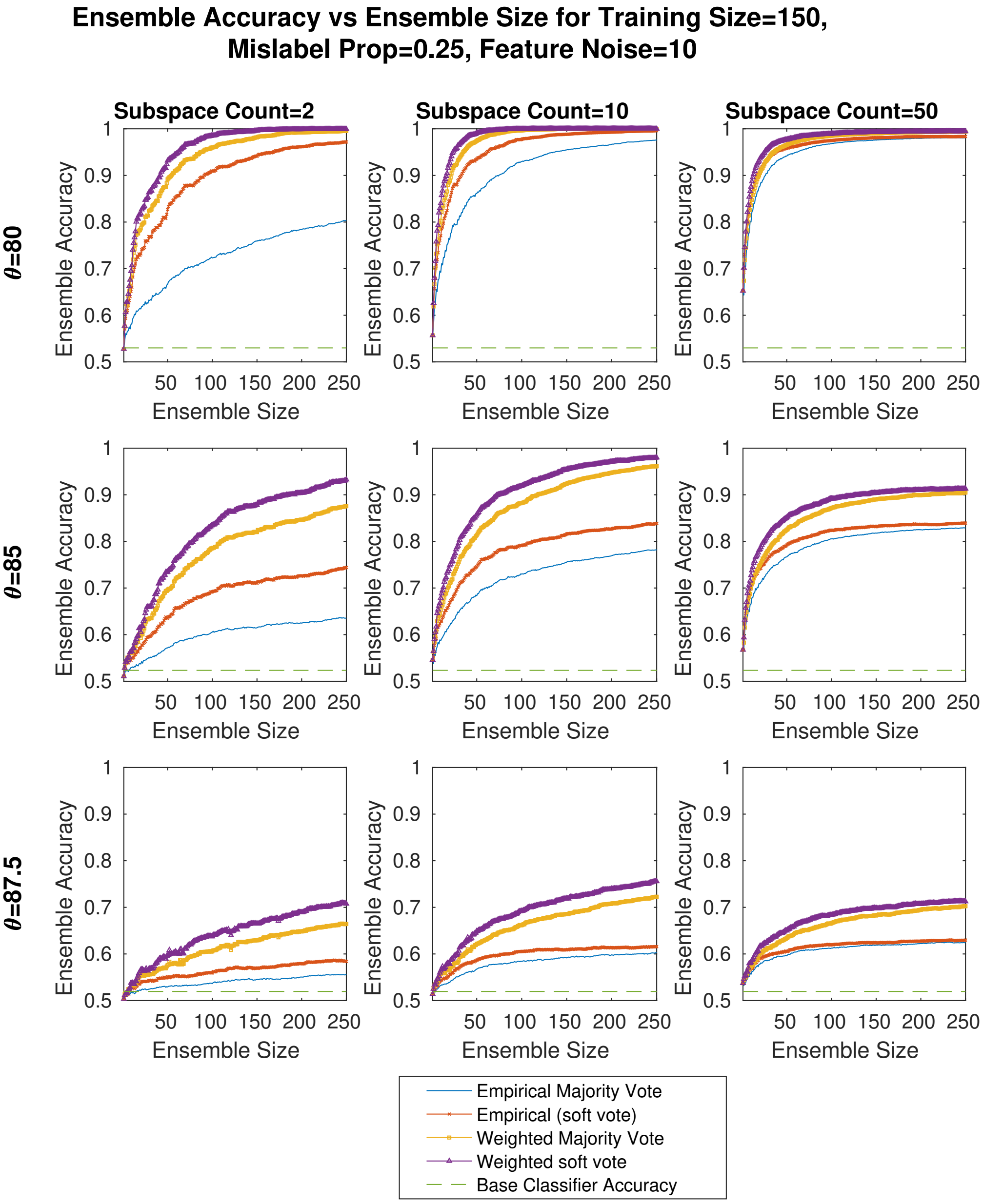

**Figure 6.7:** *Ensemble classification accuracy vs ensemble member size for a noisy setting (feature noise proportion $s = 10$, mislabel proportion $q = 0.25$), with low number of training examples $n = 150$ and dimensionality $d = 1000$*

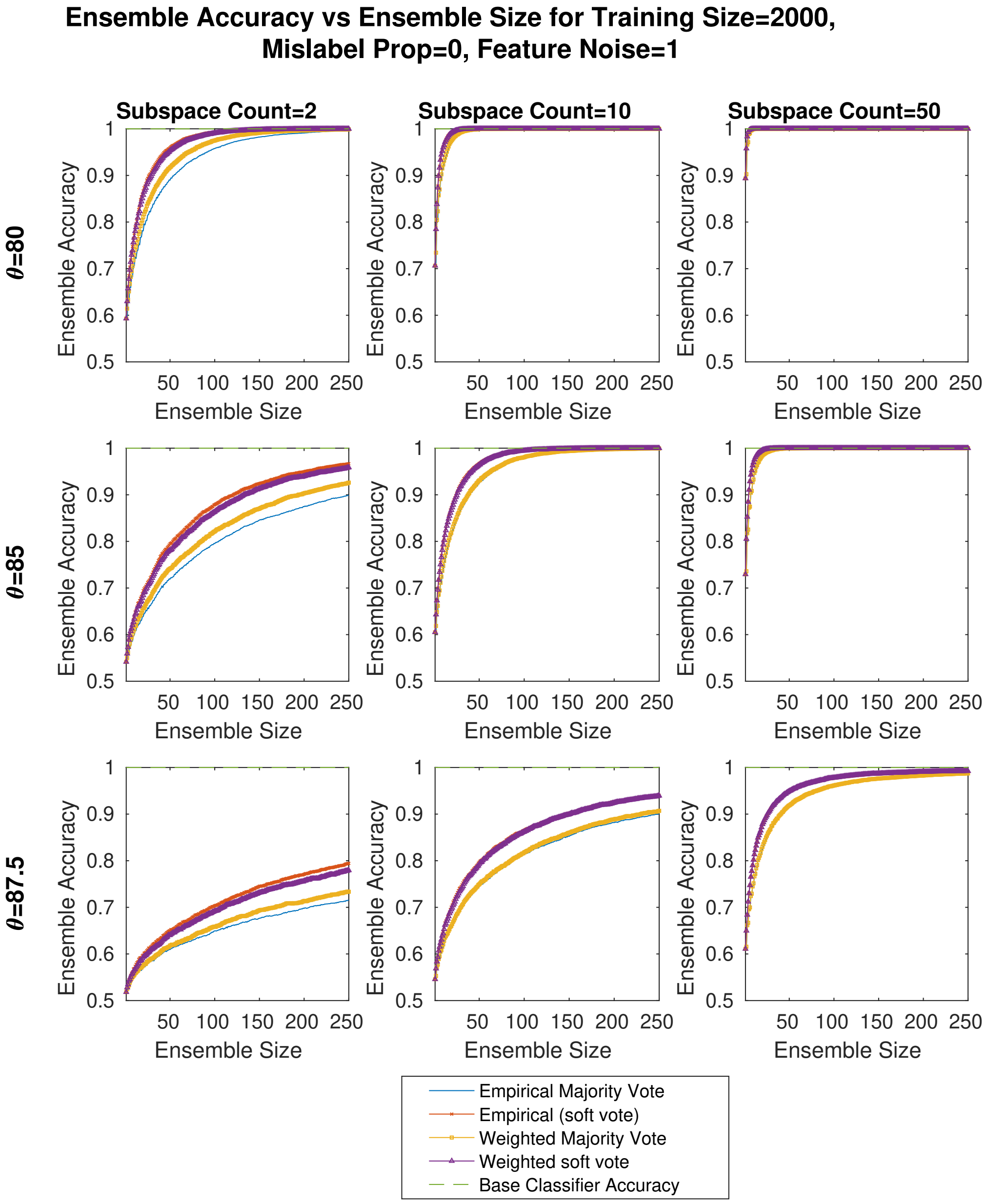

**Figure 6.8:** *Ensemble classification accuracy vs ensemble member size for a noiseless setting (feature noise proportion $s = 1$, mislabel proportion $q = 0$), with large number of training examples $n = 2000$ and dimensionality $d = 1000$*

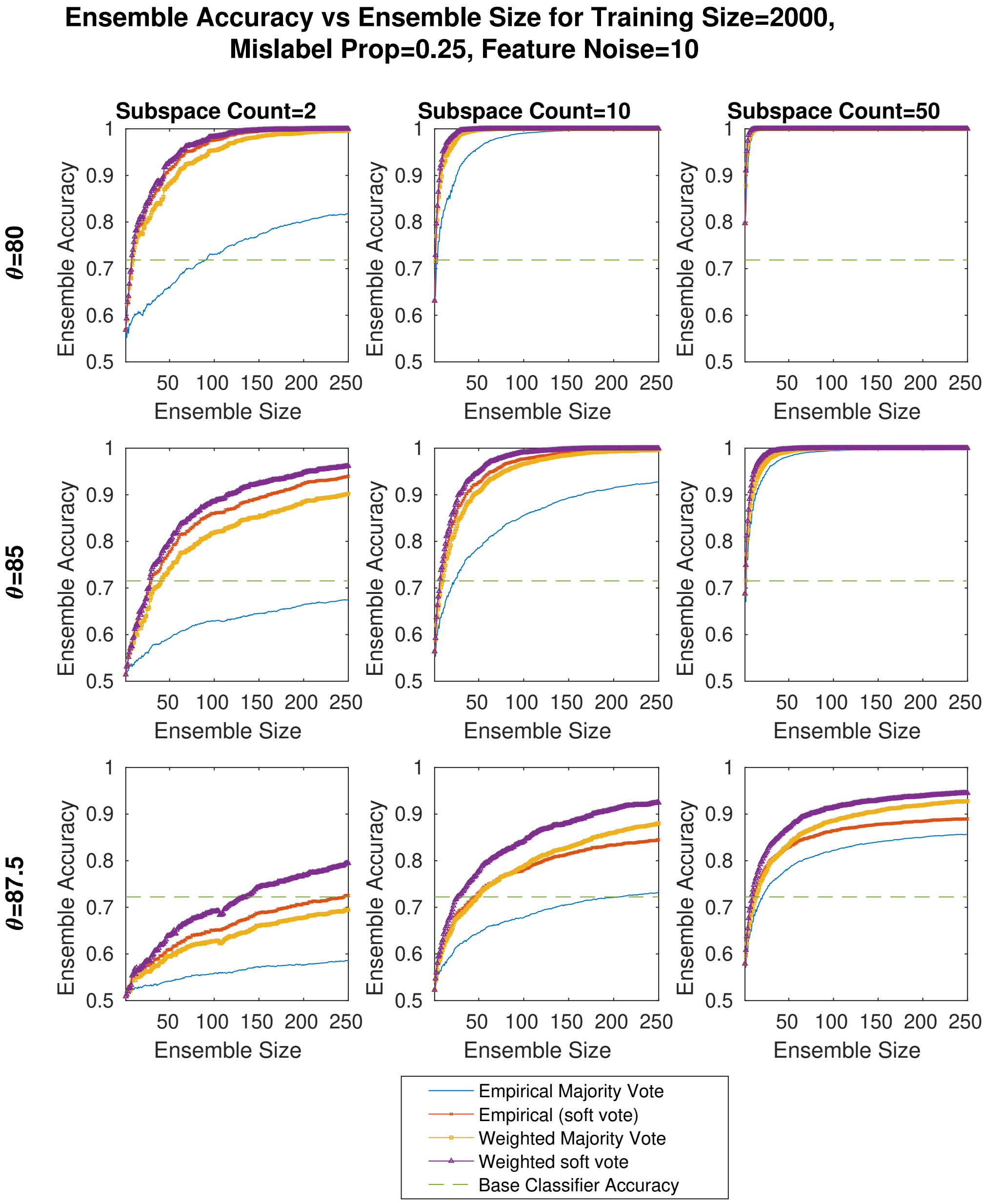

**Figure 6.9:** *Ensemble classification accuracy vs ensemble member size for a noisy setting (feature noise proportion $s = 10$, mislabel proportion $q = 0.25$), with large number of training examples $n = 2000$ and dimensionality $d = 1000$*

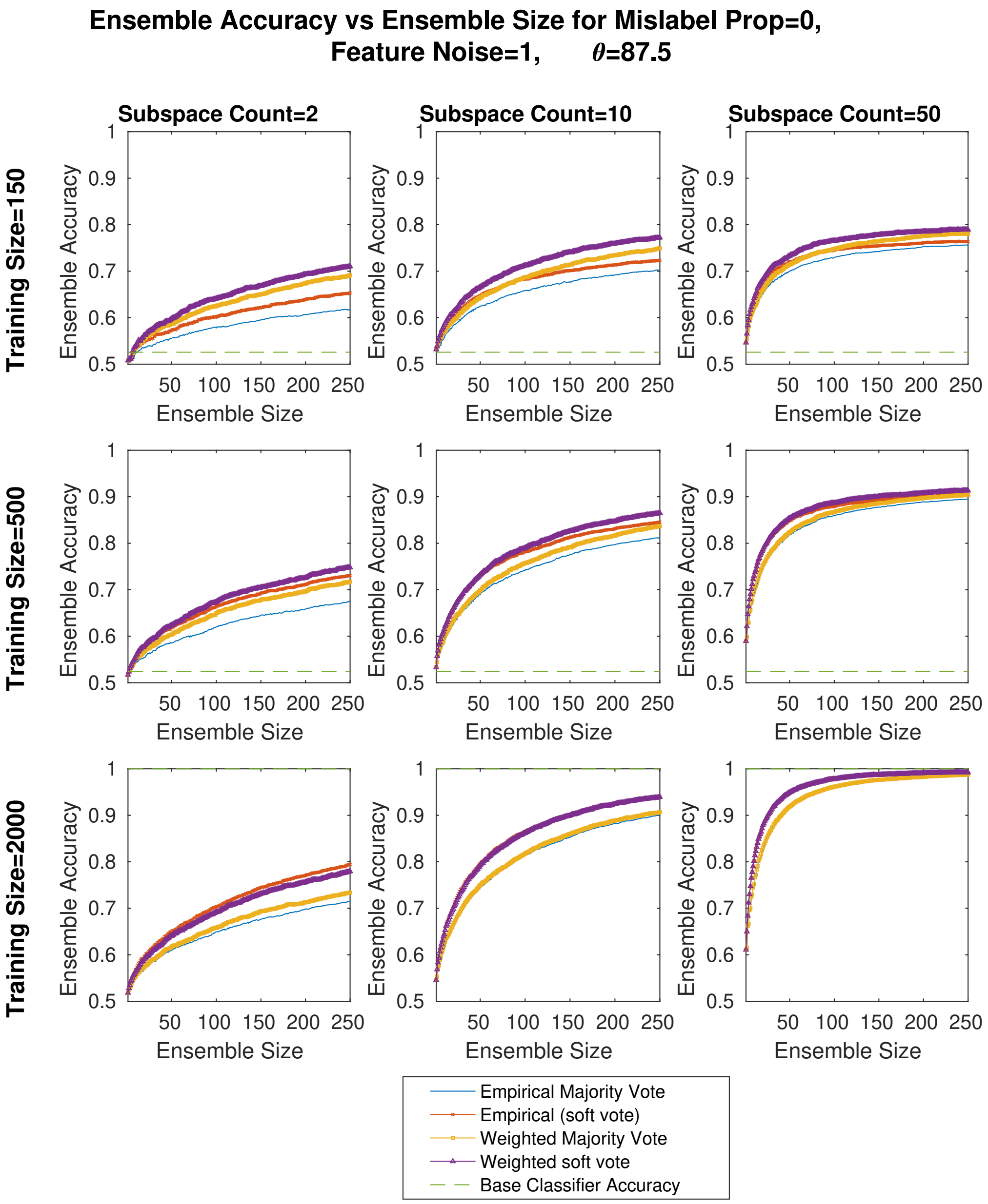

**Figure 6.10:** *Ensemble classification accuracy vs ensemble member size for a noiseless setting (feature noise proportion $s = 1$, mislabel proportion $q = 0$), on a "difficult" problem $\theta = 87.5°$. Observe here that the accuracy of the classification increases as the number of training examples provided increases.*

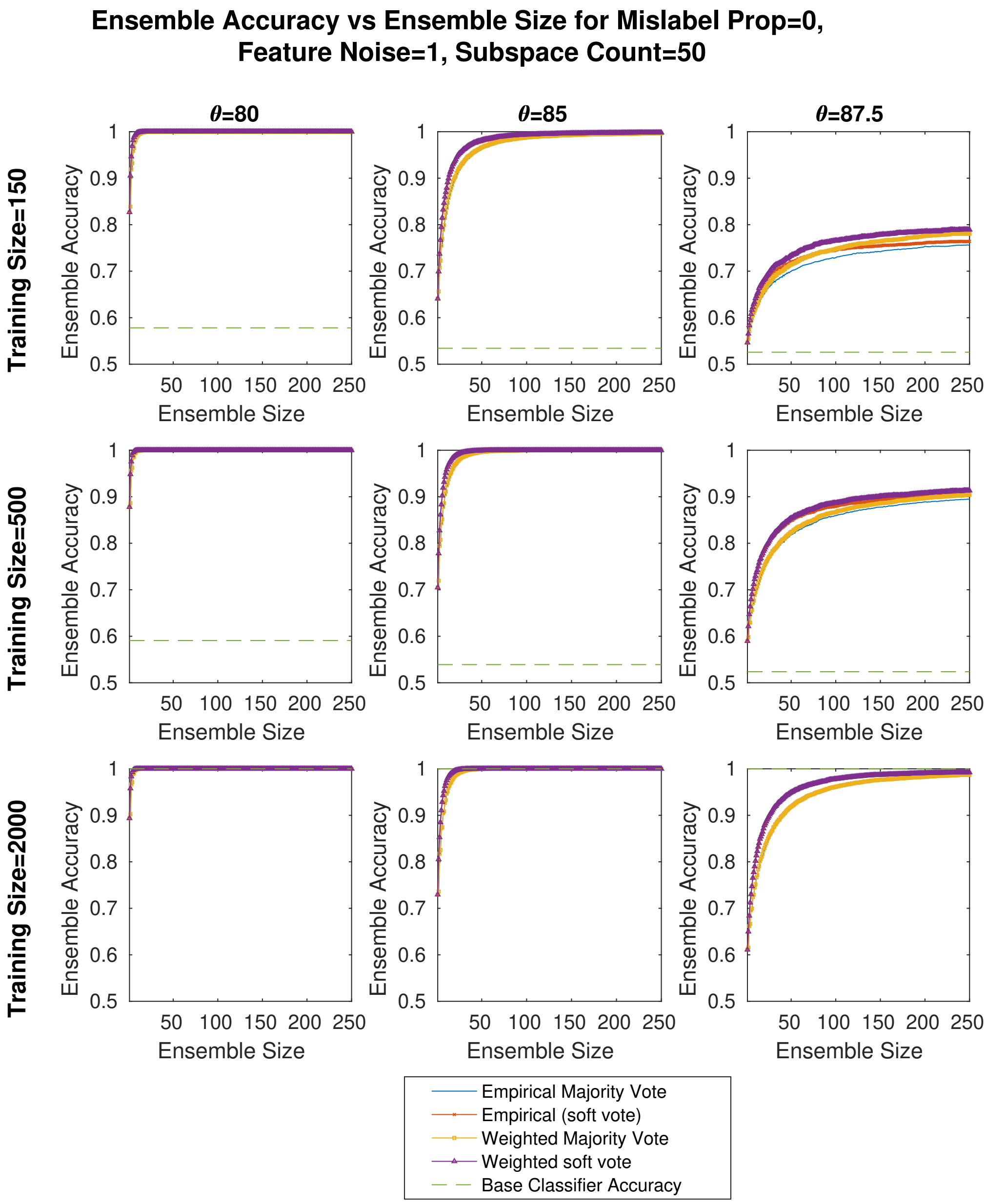

**Figure 6.11:** *Ensemble classification accuracy vs ensemble member size for a noiseless setting (feature noise proportion* $s = 1$*, mislabel proportion* $q = 0$*), with projection dimensions* $k = 50$*. Observe here that the number of training examples needed to generalize the problem increases as the angle* $\theta$ *("difficulty") increases.*

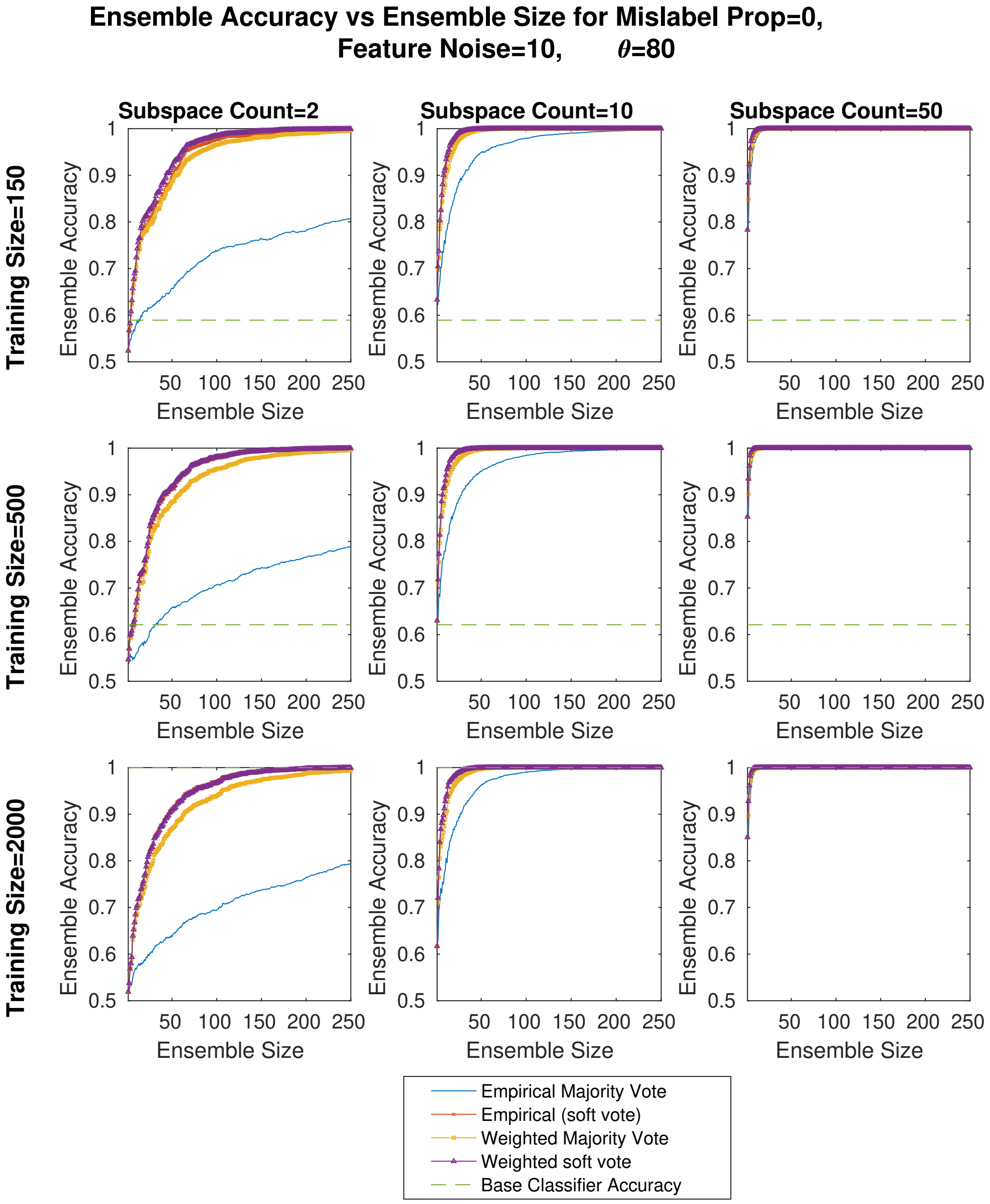

**Figure 6.12:** *Ensemble classification accuracy vs ensemble member size for a noisy setting (feature noise proportion* $s = 10$*, mislabel proportion* $q = 0$*), on an "easy problem"* $\theta = 80°$*,* $d = 1000$*. Observe that, the ensemble has poor accuracy when the number of projection dimensions is small (*$k = 2$*). This is contrasted against figure 6.13.*

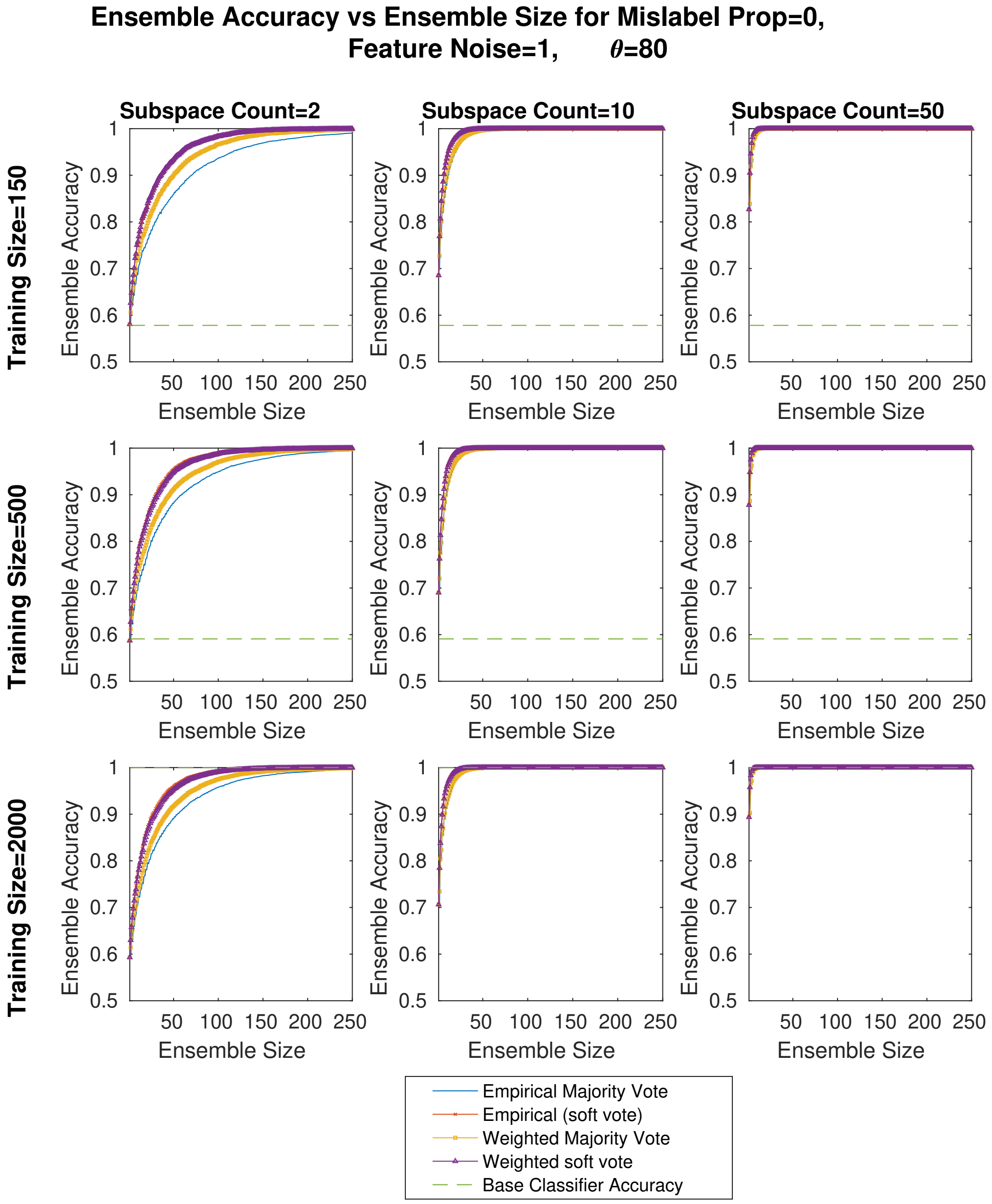

**Figure 6.13:** *Ensemble classification accuracy vs ensemble member size for a noiseless setting (feature noise proportion $s = 1$, mislabel proportion $q = 0$), on an "easy problem" $\theta = 80°$, $d = 1000$ Observe that, the ensemble has good accuracy even with a small number of projection dimensions ($k = 2$). This is contrasted against figure 6.12.*

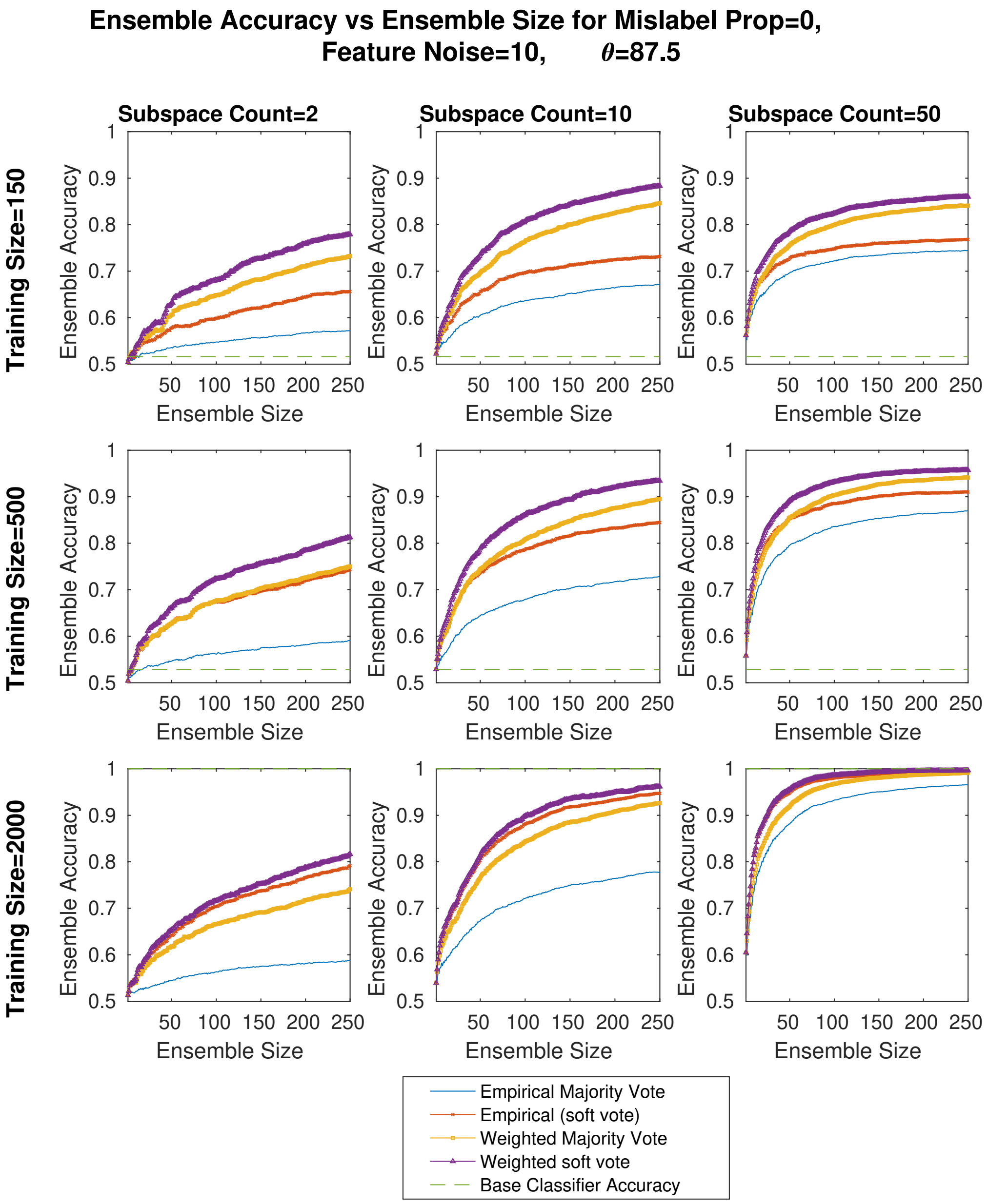

**Figure 6.14:** *Ensemble classification accuracy vs ensemble member size for a noisy setting (feature noise proportion $s = 10$, mislabel proportion $q = 0$), on a "difficult" problem, $d = 1000$. Observe here, that the sum rule classifier improves more than the other combination schemes as the number of training examples increases.*

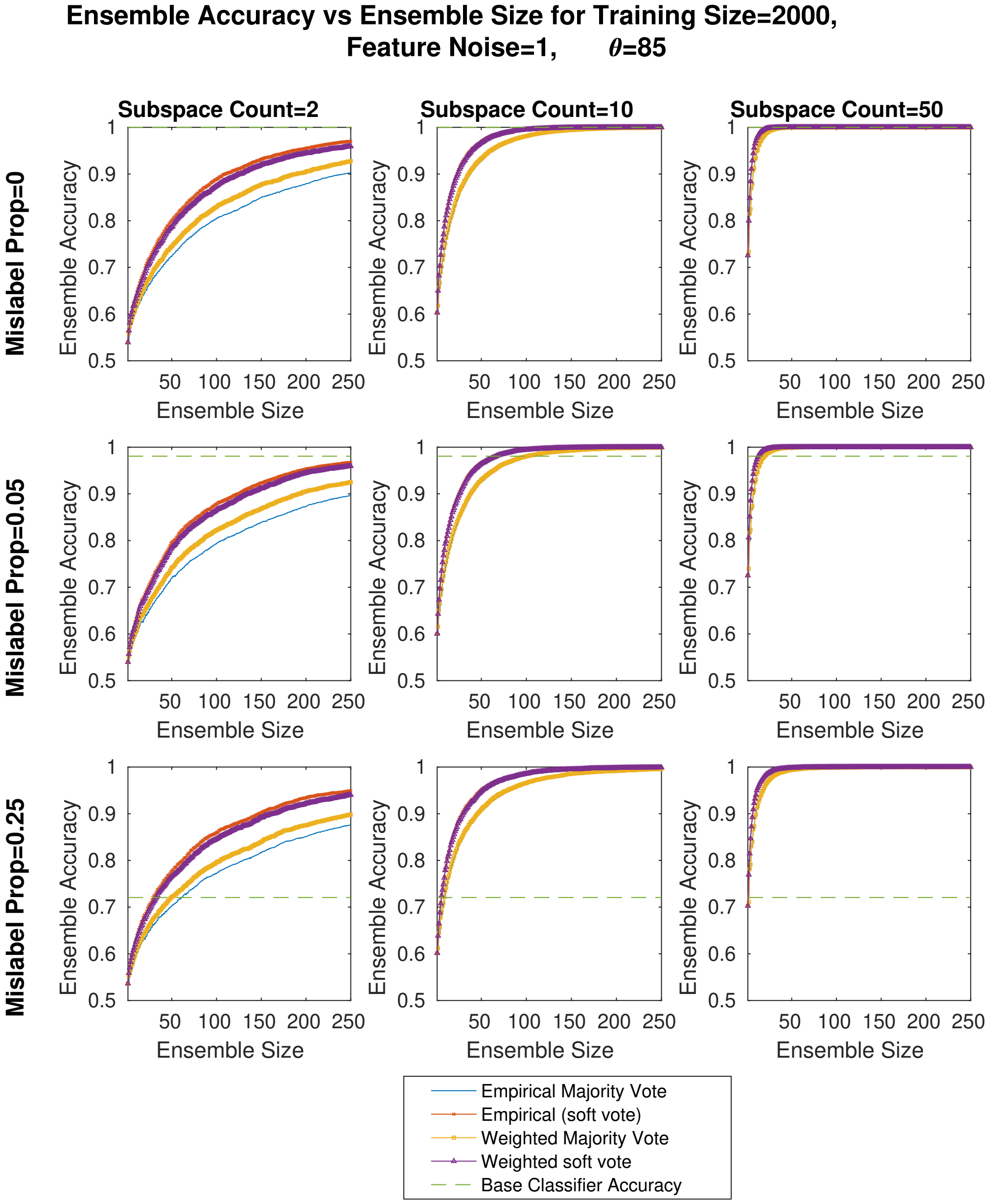

**Figure 6.15:** *Ensemble classification accuracy vs ensemble member size, with large number of training examples $n = 2000$. Here, we want to show that the classifier ensemble performance is not adversely affected by mislabelling, and that the ensemble is robust to label noise.*

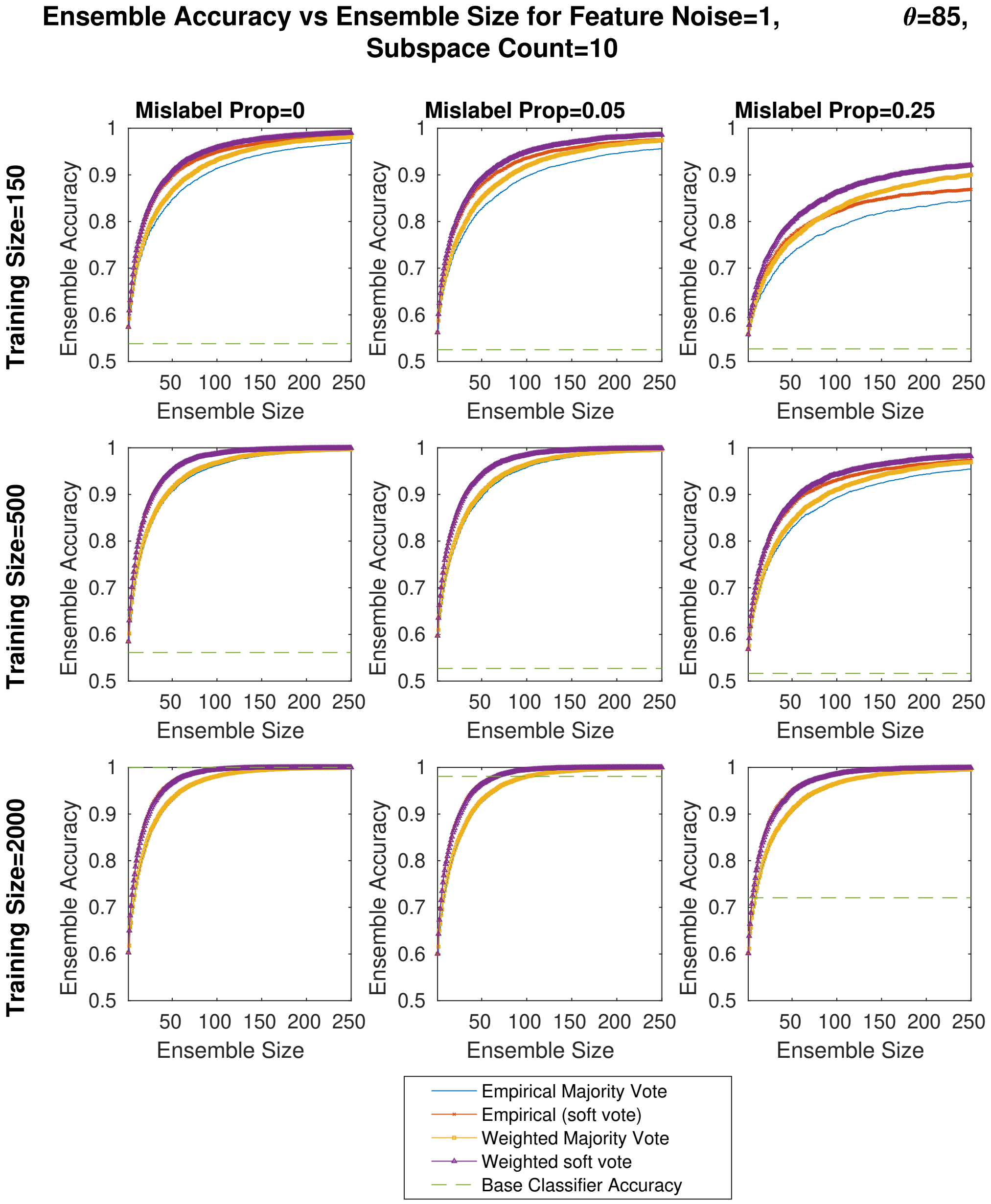

**Figure 6.16:** *Ensemble classification accuracy vs ensemble member size. Here we want to show that the robustness of the classifier ensemble to noise improves as the number of training examples increases.*

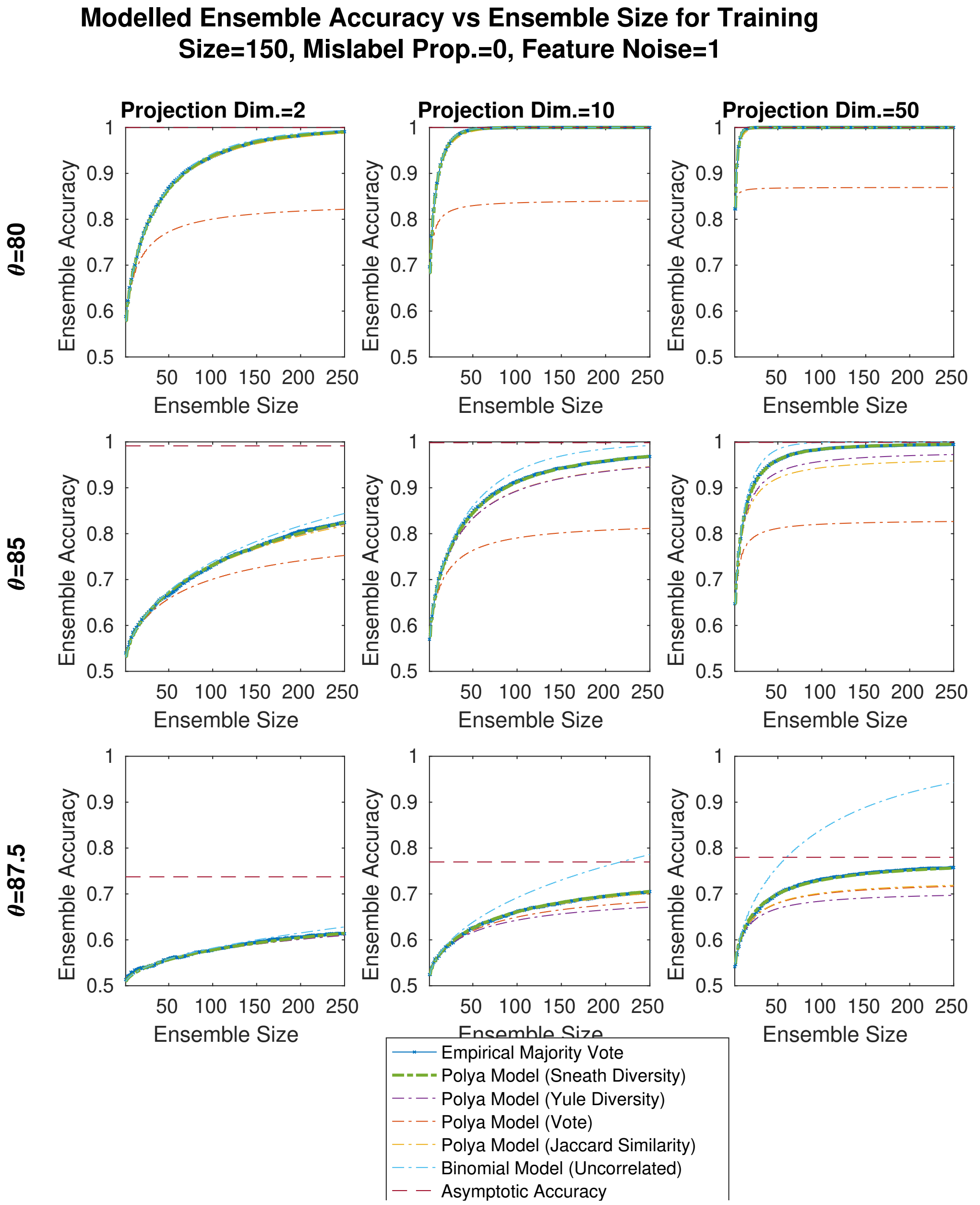

**Figure 6.17:** *Majority vote ensemble accuracy as modelled by a Polya-Eggenberger distribution vs ensemble member size for a noiseless setting ($s = 1$) and low number of training examples $n = 150$ and dimensionality $d = 1000$. Overlaid is the majority vote accuracy measured empirically. Dashed line are the accuracies as modelled by a Polya distribution model using different diversity measures. Observe that the model using the Sneath diversity measure accurately estimates the empirical majority vote accuracy.*

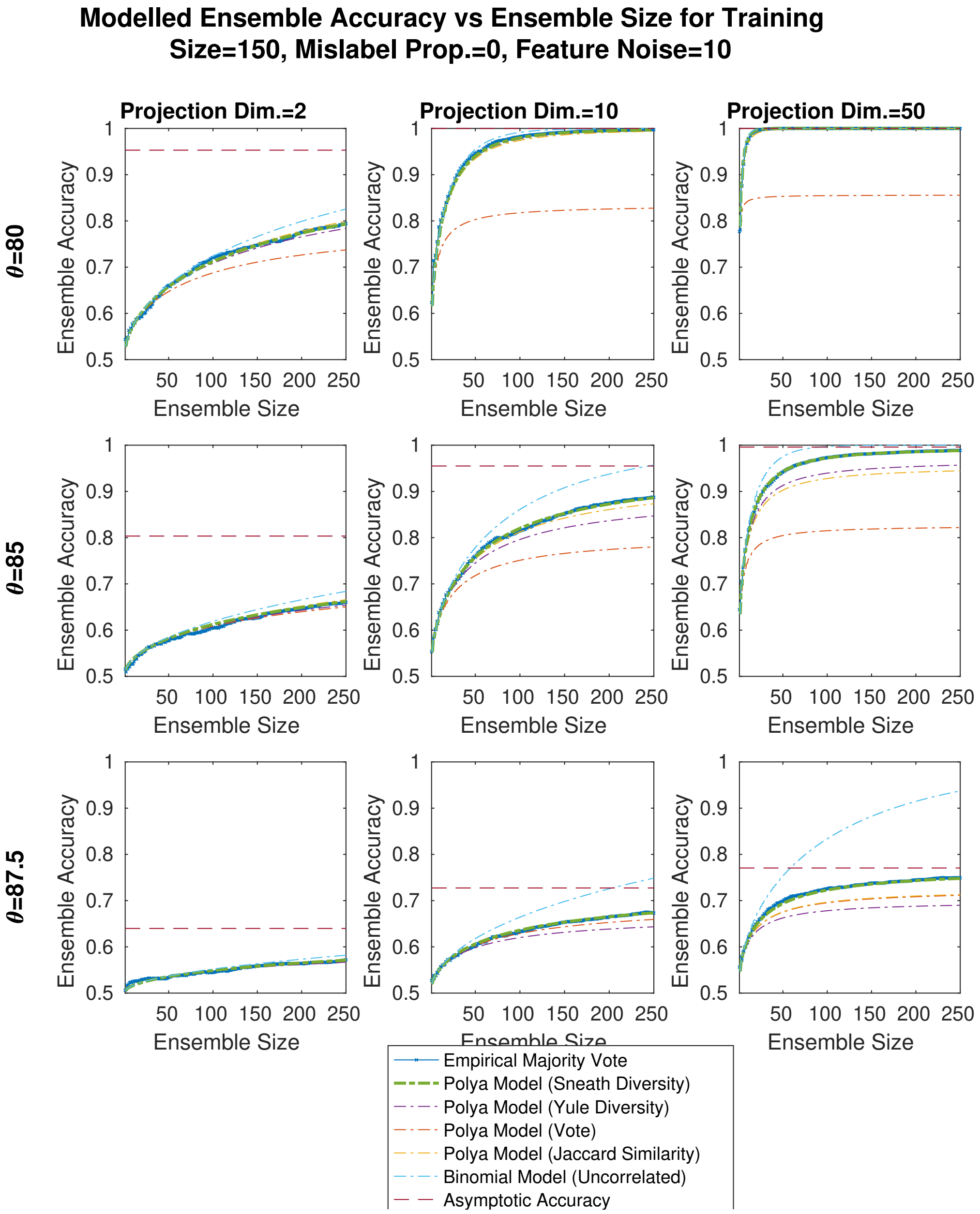

**Figure 6.18:** *Majority vote ensemble accuracy as modelled by a Polya-Eggenberger distribution vs ensemble member size for a noisy setting ($s = 10$) and low number of training examples $n = 150$ and dimensionality $d = 1000$. Overlaid is the majority vote accuracy measured empirically. Dashed line are the accuracies as modelled by a Polya distribution model using different diversity measures. Observe that the model using the Sneath diversity measure accurately estimates the empirical majority vote accuracy.*

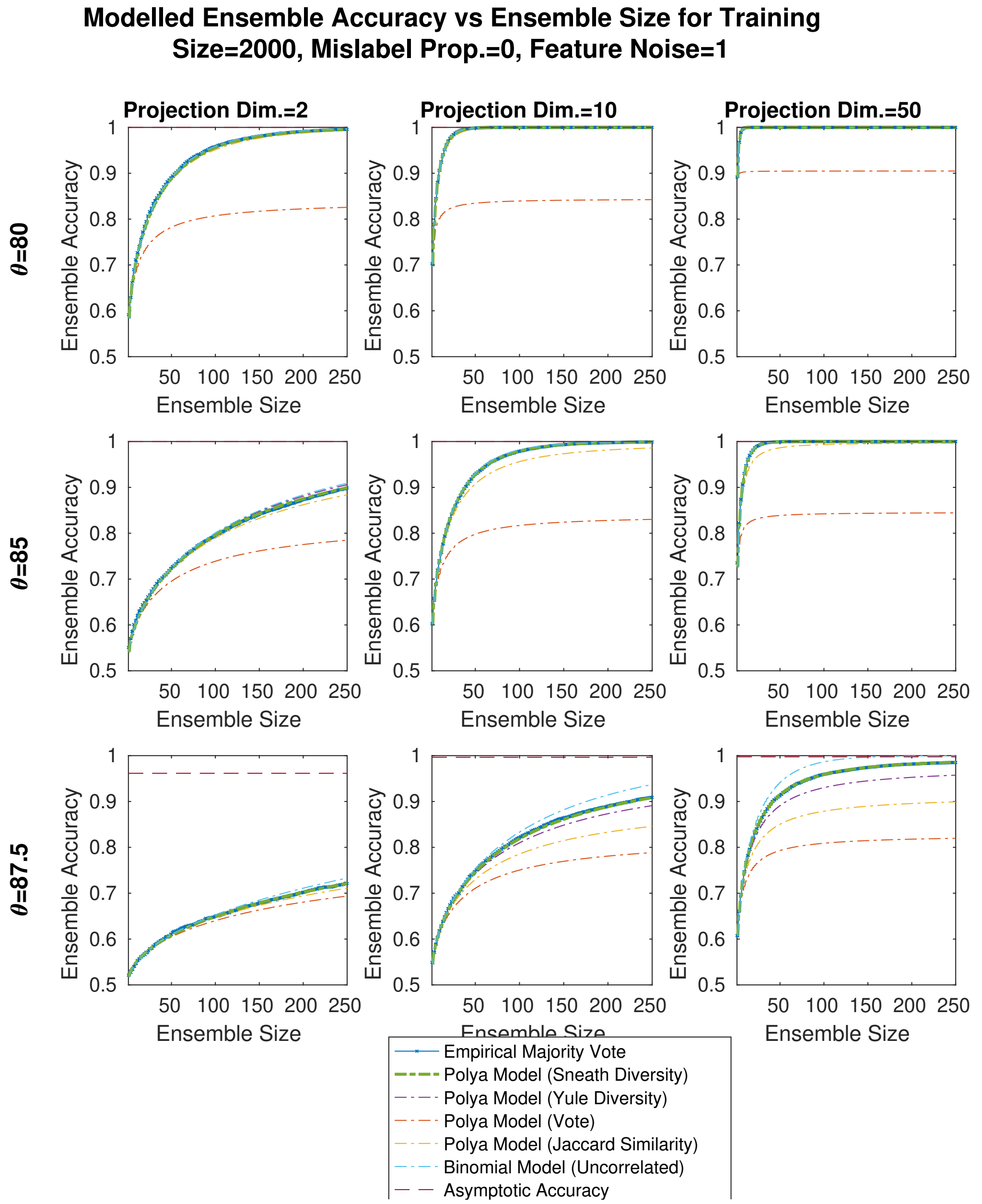

**Figure 6.19:** *Majority vote ensemble accuracy as modelled by a Polya-Eggenberger distribution vs ensemble member size for a noiseless setting* $(s = 1)$ *and large number of training examples* $n = 2000$ *and dimensionality* $d = 1000$*. Overlaid is the majority vote accuracy measured empirically. Dashed line are the accuracies as modelled by a Polya distribution model using different diversity measures. Observe that the model using the Sneath diversity measure accurately estimates the empirical majority vote accuracy.*

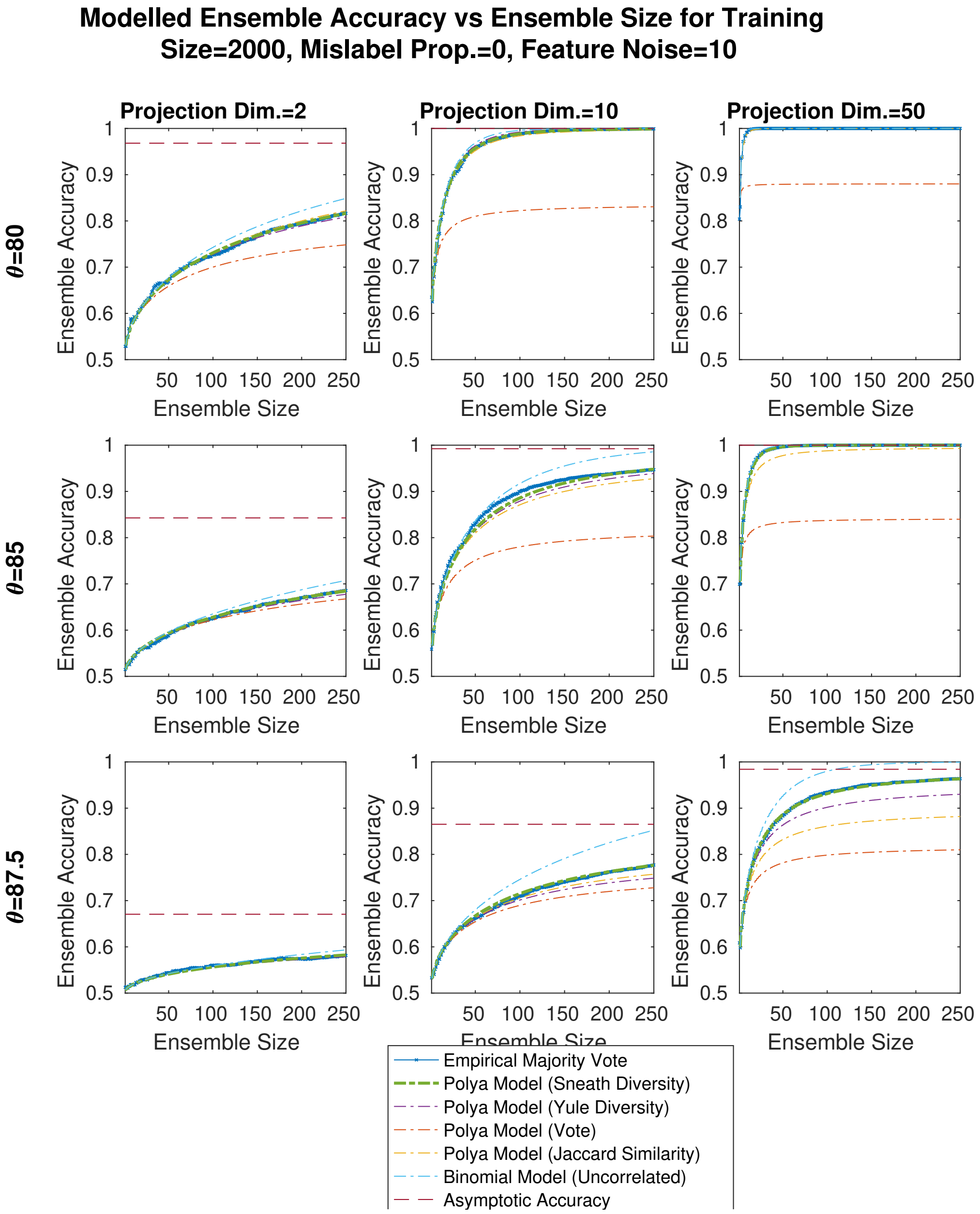

**Figure 6.20:** *Majority vote ensemble accuracy as modelled by a Polya-Eggenberger distribution vs ensemble member size for a noisy setting ($s = 10$) and large number of training examples $n = 2000$ and dimensionality $d = 1000$. Overlaid is the majority vote accuracy measured empirically. Dashed line are the accuracies as modelled by a Polya distribution model using different diversity measures. Observe that the model using the Sneath diversity measure accurately estimates the empirical majority vote accuracy.*

| | | k=2 | | | k=10 | | | k=50 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| angle | n | p | std(p) | r | p | std(p) | r | p | std(p) | r |
| 80 | 150 | 57.44 | 0.34 | 0.0005 | 66.85 | 0.32 | 0.0019 | 0.760 | 0.37 | 0.0100 |
| | 500 | 58.27 | 0.26 | 0.0005 | 69.37 | 0.29 | 0.0009 | 0.851 | 0.19 | 0.0053 |
| | 2000 | 58.55 | 0.32 | 0.0004 | 70.29 | 0.23 | 0.0004 | 0.884 | 0.17 | 0.0019 |
| 85 | 150 | 52.81 | 0.21 | 0.0009 | 56.48 | 0.22 | 0.0036 | 0.618 | 0.27 | 0.0139 |
| | 500 | 53.65 | 0.16 | 0.0005 | 58.80 | 0.22 | 0.0019 | 0.683 | 0.18 | 0.0097 |
| | 2000 | 54.11 | 0.21 | 0.0005 | 60.09 | 0.15 | 0.0008 | 0.720 | 0.20 | 0.0036 |
| 87.5 | 150 | 50.84 | 0.14 | 0.0012 | 52.00 | 0.17 | 0.0050 | 0.539 | 0.22 | 0.0188 |
| | 500 | 51.30 | 0.11 | 0.0010 | 53.17 | 0.20 | 0.0038 | 0.570 | 0.20 | 0.0186 |
| | 2000 | 51.80 | 0.10 | 0.0006 | 54.43 | 0.14 | 0.0020 | 0.602 | 0.20 | 0.0094 |

**Table 6.3:** *Parameter for Polya-Eggenberger model estimated from empirical simulation on a setting with high mislabelling and low feature noise (mislabelling proportion $q = 0.25$ and feature noise $s = 1$). Values of p is given in percentage.*

| | | k=2 | | | k=10 | | | k=50 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| angle | n | p | std(p) | r | p | std(p) | r | p | std(p) | r |
| 80 | 150 | 53.15 | 0.54 | 0.0012 | 61.74 | 1.02 | 0.0036 | 78.90 | 0.84 | 0.0058 |
| | 500 | 53.25 | 0.43 | 0.0012 | 62.32 | 0.87 | 0.0036 | 83.65 | 0.84 | 0.0059 |
| | 2000 | 53.15 | 0.54 | 0.0013 | 63.10 | 0.62 | 0.0034 | 85.02 | 1.00 | 0.0061 |
| 85 | 150 | 51.69 | 0.34 | 0.0012 | 55.62 | 0.46 | 0.0041 | 63.69 | 0.49 | 0.0108 |
| | 500 | 51.73 | 0.29 | 0.0013 | 56.39 | 0.53 | 0.0036 | 68.62 | 0.66 | 0.0084 |
| | 2000 | 51.63 | 0.30 | 0.0012 | 56.98 | 0.54 | 0.0032 | 70.46 | 0.81 | 0.0057 |
| 87.5 | 150 | 50.70 | 0.19 | 0.0013 | 52.14 | 0.25 | 0.0050 | 54.77 | 0.32 | 0.0168 |
| | 500 | 50.80 | 0.17 | 0.0012 | 52.77 | 0.28 | 0.0044 | 57.80 | 0.23 | 0.0157 |
| | 2000 | 50.77 | 0.19 | 0.0012 | 53.26 | 0.22 | 0.0037 | 59.93 | 0.46 | 0.0094 |

**Table 6.4:** *Parameter for Polya-Eggenberger model estimated from empirical simulation on a setting with high number of irrelevant features (mislabelling proportion $q = 0$ and feature noise $s = 10$). Values of p is given in percentage. Observe that the variation in p is larger than when the feature noise is small.*

|  |  | k=2 |  |  | k=10 |  |  | k=50 |  |  |
|---|---|---|---|---|---|---|---|---|---|---|
| angle | n | p | std(p) | r | p | std(p) | r | p | std(p) | r |
| 80 | 150 | 53.24 | 0.50 | 0.0012 | 61.01 | 0.91 | 0.0038 | 72.86 | 0.79 | 0.0110 |
|  | 500 | 53.31 | 0.57 | 0.0012 | 62.37 | 1.19 | 0.0036 | 81.21 | 0.92 | 0.0083 |
|  | 2000 | 53.05 | 0.55 | 0.0012 | 62.90 | 0.91 | 0.0033 | 84.42 | 0.84 | 0.0065 |
| 85 | 150 | 51.52 | 0.31 | 0.0012 | 55.06 | 0.50 | 0.0044 | 61.02 | 0.49 | 0.0146 |
|  | 500 | 51.62 | 0.24 | 0.0012 | 56.09 | 0.56 | 0.0040 | 66.60 | 0.39 | 0.0123 |
|  | 2000 | 51.63 | 0.29 | 0.0012 | 56.70 | 0.64 | 0.0034 | 70.00 | 0.56 | 0.0070 |
| 87.5 | 150 | 50.60 | 0.15 | 0.0012 | 51.73 | 0.19 | 0.0052 | 53.70 | 0.26 | 0.0190 |
|  | 500 | 50.78 | 0.17 | 0.0012 | 52.58 | 0.35 | 0.0046 | 56.52 | 0.40 | 0.0194 |
|  | 2000 | 50.82 | 0.15 | 0.0012 | 53.22 | 0.29 | 0.0039 | 59.44 | 0.30 | 0.0120 |

**Table 6.5:** *Parameter for Polya-Eggenberger model estimated from empirical simulation on a noisy setting (mislabelling proportion $q = 0.25$ and feature noise $s = 1$. Values of p is given in percentage. Observe that the variation in p is larger than when the feature noise is small.*

|  |  | k=2 |  |  |  |
|---|---|---|---|---|---|
| theta | n | N=50 | N=100 | N=150 | N=250 |
| 80 | 150 | 66.0 / 66.6 | 71.4 / 72.3 | 75.0 / 76.0 | 80.7 / 80.7 |
|  | 500 | 66.7 / 66.9 | 73.4 / 72.7 | 76.7 / 76.4 | 81.4 / 81.1 |
|  | 2000 | 67.1 / 66.9 | 73.6 / 72.6 | 76.5 / 76.4 | 81.1 / 81.1 |
| 85 | 150 | 58.6 / 58.6 | 61.6 / 61.8 | 64.3 / 64.0 | 66.9 / 67.0 |
|  | 500 | 59.1 / 59.3 | 61.8 / 62.7 | 64.2 / 65.0 | 68.4 / 68.3 |
|  | 2000 | 60.0 / 59.9 | 63.5 / 63.5 | 65.8 / 66.0 | 69.4 / 69.5 |
| 87.5 | 150 | 53.6 / 53.6 | 55.0 / 55.0 | 56.3 / 55.9 | 57.2 / 57.2 |
|  | 500 | 53.9 / 54.0 | 55.6 / 55.5 | 56.9 / 56.6 | 58.3 / 58.1 |
|  | 2000 | 54.3 / 54.4 | 56.2 / 56.1 | 57.4 / 57.3 | 58.8 / 58.9 |

**Table 6.6:** *Comparison between the empirical majority vote ensemble accuracy (left) and our model (right) for data with many irrelevant features. Observe that the values are within 1% of the model.*

|  |  | k=2 |  |  |  |
|---|---|---|---|---|---|
| theta | n | N=50 | N=100 | N=150 | N=250 |
| 80 | 150 | 85.0 / 86.0 | 93.7 / 93.5 | 96.9 / 96.7 | 99.0 / 99.0 |
|  | 500 | 88.3 / 88.2 | 95.1 / 95.2 | 97.7 / 97.8 | 99.5 / 99.5 |
|  | 2000 | 88.7 / 88.9 | 96.0 / 95.7 | 98.3 / 98.2 | 99.6 / 99.6 |
| 85 | 150 | 67.6 / 66.8 | 72.8 / 72.7 | 77.0 / 76.6 | 81.7 / 81.7 |
|  | 500 | 69.9 / 70.3 | 76.9 / 77.2 | 81.6 / 81.7 | 87.2 / 87.2 |
|  | 2000 | 72.3 / 72.5 | 80.4 / 79.9 | 85.0 / 84.6 | 90.3 / 90.2 |
| 87.5 | 150 | 55.4 / 55.5 | 57.6 / 57.6 | 59.1 / 59.1 | 61.1 / 61.2 |
|  | 500 | 58.5 / 58.3 | 61.2 / 61.5 | 63.5 / 63.8 | 66.9 / 67.0 |
|  | 2000 | 60.5 / 60.6 | 64.1 / 64.7 | 67.2 / 67.6 | 71.8 / 71.8 |

**Table 6.7:** *Comparison between the empirical majority vote ensemble accuracy (left) and our model (right) for data with no uninformative features. Observe that the values are within* 1% *of the model.*

on the dataset before classifying the datasets using LDA. We then used a majority vote ensemble and measured the ensemble accuracy versus ensemble member size a various number of subspaces. As a reference, we plotted the base classifier (LDA with all the features) when possible, as well as the various ensemble combination schemes, namely, weighted majority vote, soft vote and weighted soft vote.

### 6.6.1 Discussion

The NIPS feature challenge datasets comprise of six datasets that are distinct, and here we can see how these distinctiveness results in the different RS ensemble accuracy performance,

The ARCENE data is dense, and therefore, we expect RS ensembles to work well on this data in accordance to our results and theory from chapter 4 and 5. Here we observed in figure 6.21 that weighing the majority vote and sum rule ensemble gives very little improvement on the accuracy of the ensemble. This also indicates to us that most of the features in this dataset is informative and there is very little feature noise. This observation is consistent with our synthetic results in which we have very similar accuracies for both majority vote and sum rule combination schemes. We expect some generalization error due to the small sample size in comparison to the dimensionality of the data. We also observed that the accuracies plateau very quickly indicating to us that many of the features might be correlated to each other. In figure 6.22, interestingly we see that the vote correlation appears to approximate the diversity measure $r$ quite well for some of subspace counts ($k = 50$ and $k = 75$). However, this may just be a coincidence. Finally, we also observe that the ensembles with extremely low subspace counts ($k \leq 2$) gives us an ensemble with very poor ensemble accuracy performance, exhibiting the same non-monotonic majority vote accuracy behaviour as our simulation in section 5.4 (see figure 5.9), leading credence that the number of subspaces are too small for the problem resulting in inconsistent classifier behaviour, in line with our results from chapters 4 and 5.

DEXTER is sparser compared to ARCENE with a large number of irrelevant features. We see in figure 6.23 the results are consistent with our theory and observations with synthetic data. First, using a weighted combination scheme results in a substantial accuracy gain over an unweighted combination scheme. Second, also consistent with our theory, the number of subspaces in the classifiers required for DEXTER to give consistent ensemble performance is higher than in the case of ARCENE, (namely $k \geq 100$). We also see in figure 6.24, that when the number of projection dimensions is small ($k < 100$), we have very poor majority vote accuracy and inconsistent ensemble behaviour. This is similar to what we observed in ARCENE when $k \leq 2$ and is consistent with our expectation DEXTER will require a larger number of subspaces in comparison to ARCENE. This again is consistent with our result in chapter 4, where DEXTER having a substantially larger regularity constant $c'$ would also require substantially larger number of subspaces than ARCENE in order to give consistent classifier performance. Interestingly, we also observed that for the classifier subspace count $k = 200$, the soft-voting ensemble had lower accuracy than that of a majority vote, indicating to us that the classes in DEXTER may not be linearly separable.

DOROTHEA is extremely sparse, and according to our theory, the data is challenging for RS ensembles as clearly seen in figure 6.25. As such, we expect very poor accuracies for the RS ensembles. We do note that increasing the subspaces does improve the accuracy; however, the overall accuracy is poor compared to the other datasets and an RS ensemble may not be suitable classifier for DOROTHEA.

We observed that no choice of the size of subspace result in a consistent ensemble for DOROTHEA. As noted in chapter 4, DOROTHEA having a large regularity constant $c'$, would be a challenging classification problem for RS ensemble classifiers.

GISETTE is very similar to ARCENE, in that the data is also dense and has the smallest regularity constant $c'$ among the datasets. As such, according to our theory, we would expect that an RS ensemble to work well. This is reflected in our results as shown in figure 6.26. Similar to ARCENE, we observed that weighting the majority vote/sum rule ensemble gives little improvement on the accuracy of the ensemble. This also indicates to us that most of the features in this dataset is informative and there is very little feature noise in the data. Like ARCENE, we have very similar accuracies for both the majority vote and sum rule ensembles. In Figure 6.27, we see consistent classifier behaviour even for small number of subspaces. This is consistent with our expectation for dense datasets with large number of training samples. Overall, the RS ensembles outperformed the 'strong' base classifier, which uses all the features in the data, and our model accurately predict the majority vote classifier ensemble accuracies.

MADELON is synthetic data with the class labels determined using an XOR decision boundary over a few features (Guyon, 2003), and many non-informative features added into the data. As such, we can see in figure 6.28, the linear classifiers have difficulty classifying MADELON with a good level of accuracy. Unlike the other datasets, MADELON is the only dataset that gives better accuracies when the subspace counts are low, with the highest accuracies when the number of projection dimension, $k = 1$ where the classification task on MADELON gives the best accuracies on weighted majority vote ensembles with a small projection dimensions. Consistent with our theory, we also see that the weighted majority outperforms both the weighted and unweighted sum rule ensembles. This is because the sum rule ensembles will only approximate a linear classifier when combined as an ensemble. The majority vote ensemble on the other hand, adds non-linearity to the ensemble allowing the ensemble to better approximate the decision boundaries. Due to the contradicting expectations, an ensemble approach using RS would not be suitable for MADELON.

| File | Type | Training Set | Validation Set | Number features | Regularity Constant (c’) |
|---|---|---|---|---|---|
| ARCENE | Non sparse | 100 | 100 | 10000 | 11.44 |
| DEXTER | Sparse integer | 300 | 300 | 20000 | 132.10 |
| DOROTHEA | Sparse binary | 800 | 350 | 100000 | 32.51 |
| GISETTE | Non sparse | 6000 | 1000 | 5000 | 10.76 |
| MADELON | Non sparse | 2000 | 600 | 500 | 21.45 |

**Table 6.8:** *Summary of the characteristics of the UCI datasets.*

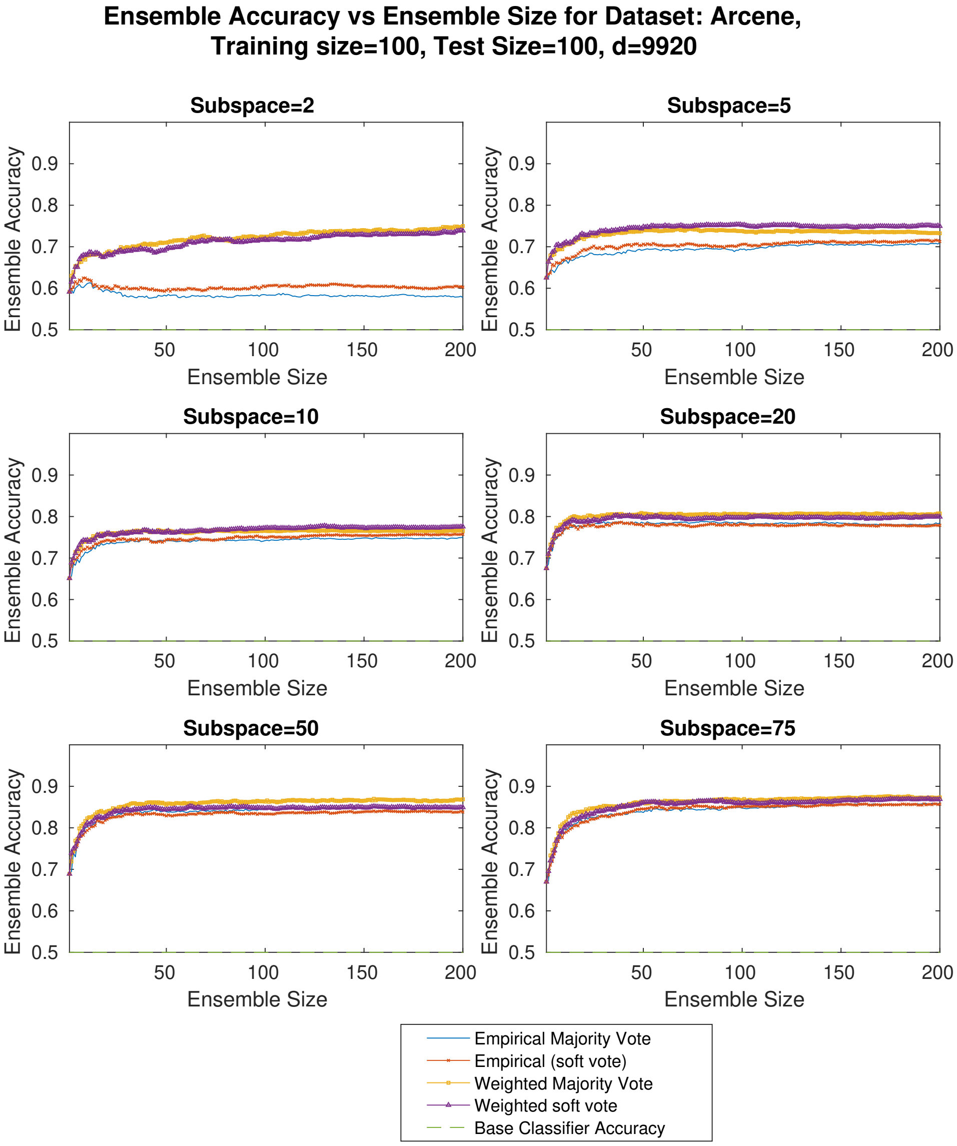

**Figure 6.21:** *Classification accuracy vs ensemble member size for ARCENE. Observe that the ensemble fails to produce a consistent majority vote ensemble when the number of projection dimensions is small, similar the simulation in chapter 5.*

## 6.7 Conclusion and Summary

In this chapter, we showed that the accuracy of a majority vote ensemble could be effectively modelled using a Polya-Eggenberger model. We also showed that the parameters of this model can be estimated from the Sneath and Sokal (1963) correlation measure ($\rho$). We discussed the implications of the model and showed how we can estimate the majority vote ensemble accuracy. We evaluated other methods of estimating the diversity measure and the limitations and shortcomings of using those approaches. We also showed that the accuracy of a soft-vote (sum-rule) ensemble improves asymptotically with the ensemble size under some mild conditions.

We corroborated our theories through extensive empirical simulations, using both synthetic data and real-world data from the NIPS feature challenge and showed how our theory accurately predicted the performance of RS ensembles on these data.

We also reconciled the findings of Schapire (1990) and Blum (1997) against Kuncheva and Rodríguez (2014) and showed that a weighted scheme improves the accuracy of the classification ensemble more . We demonstrated that a weighted scheme on a dataset with high feature noise could give improved ensemble performance. Similar to the results by Durrant and Kabán (2014), we showed that RS can work as regularization in linear classifiers ensembles but with weaker performance compared to RP ensembles.

In the next chapter, we will apply the intuitions from our theory on an image classification task and show how we can improve the image classification performance of pretrained Deep Neural Networks without retraining the network using a simple ensemble approach with random subspaces.

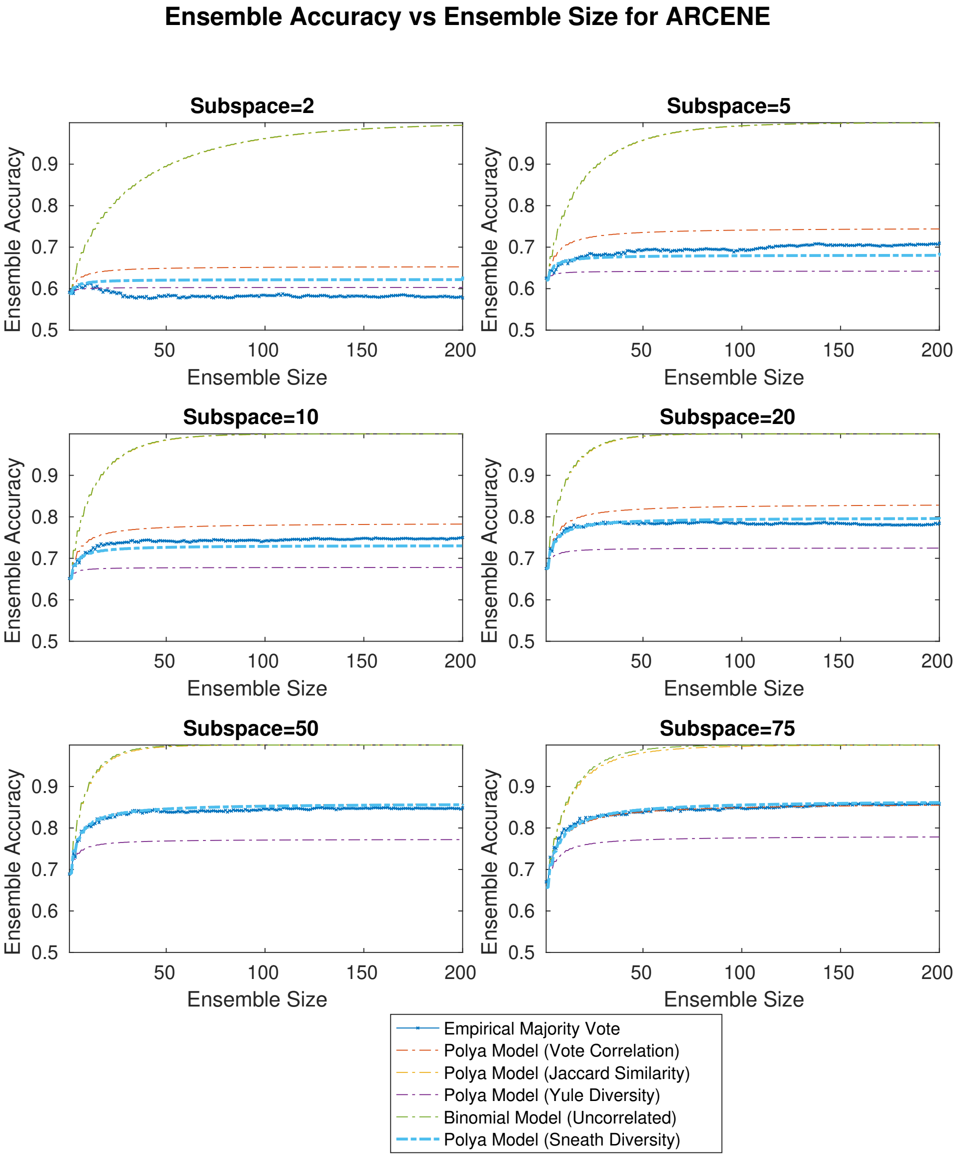

**Figure 6.22:** *Majority vote ensemble accuracy as modelled by a Polya-Eggenberger distribution vs ensemble member size for ARCENE. Observe that the ensemble fails to produce a consistent majority vote ensemble when the number of projection dimensions is small, similar the simulation in chapter 5. Observe also that for a projection dimension $k$ at least 2, our model accurately estimates the majority vote accuracy of the ensemble.*

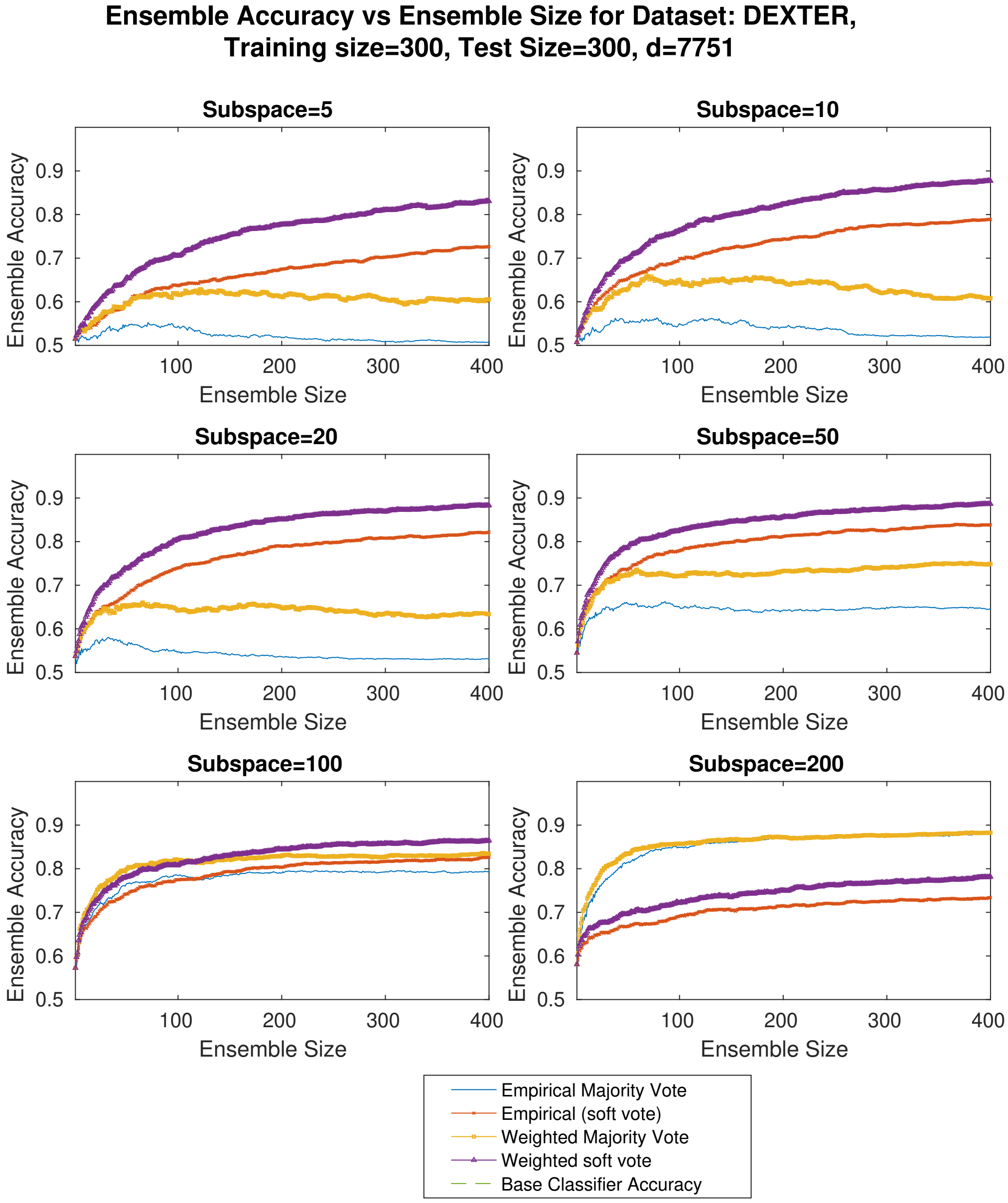

**Figure 6.23:** *Classification accuracy vs ensemble member size for DEXTER. Observe that the ensemble classifier has difficulty on DEXTER when the number of projection dimensions is less than 100.*

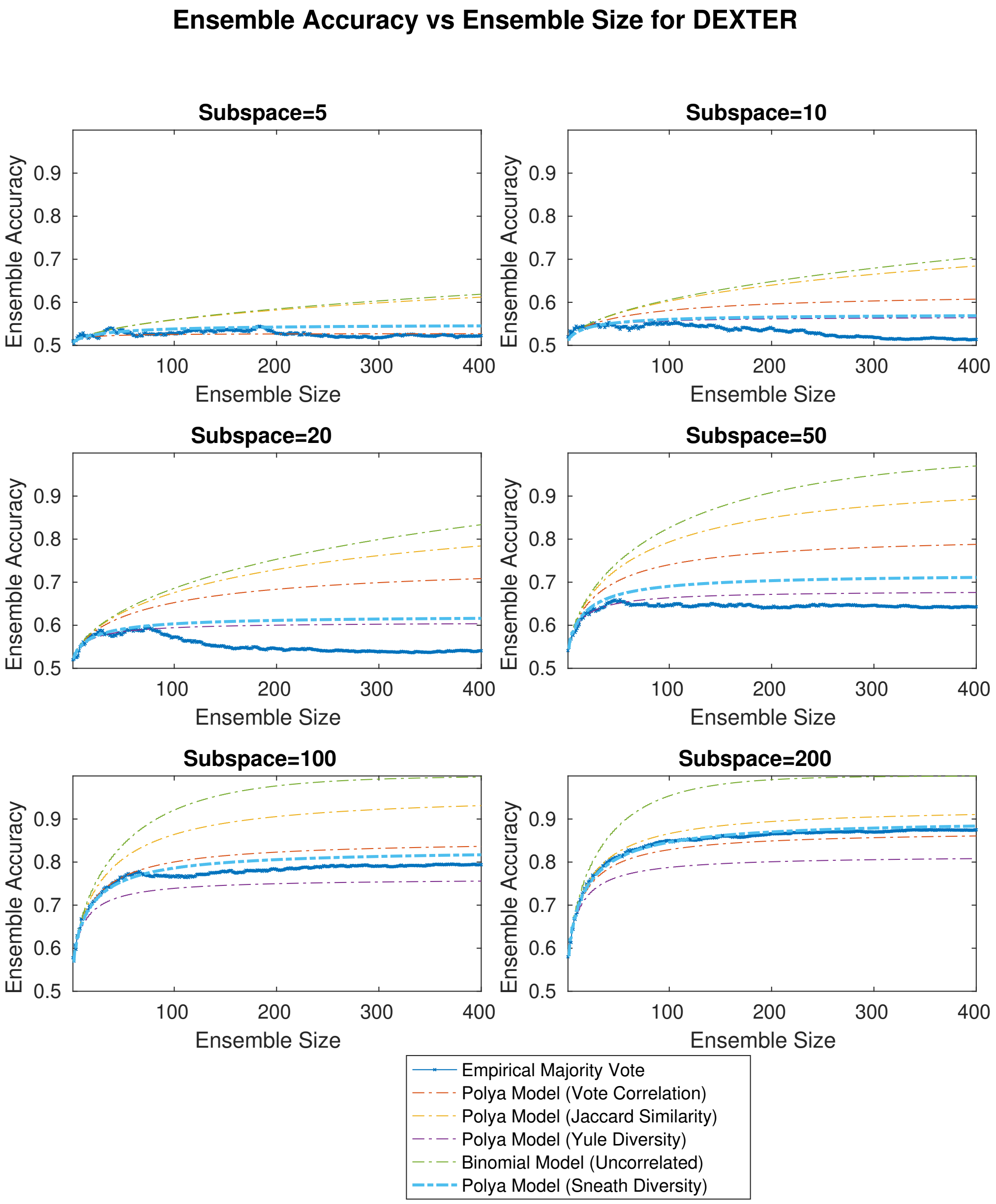

**Figure 6.24:** *Majority vote ensemble accuracy as modelled by a Polya-Eggenberger distribution vs ensemble member size for DEXTER. Observe that the ensemble requires at least a projection dimension of at least 100 to produce a consistent majority vote ensemble. Observe also that for a projection dimension at least 100, our model accurately estimates the majority vote accuracy of the ensemble.*

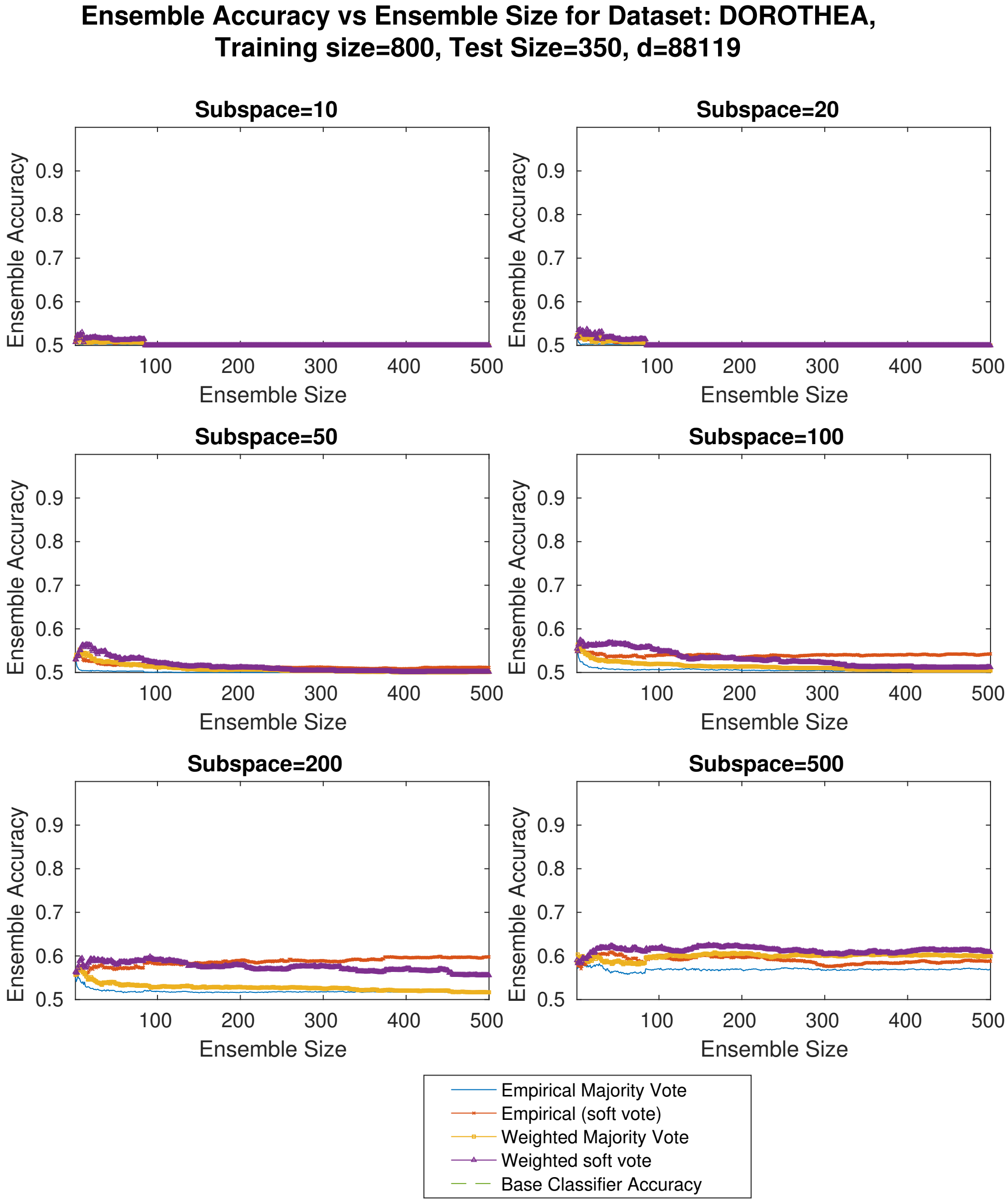

**Figure 6.25:** *Balanced classification accuracy vs ensemble member size for DOROTHEA. Observe that the ensemble classifier has difficulty on DOROTHEA, regardless of the number of projection dimensions in the classifiers.*

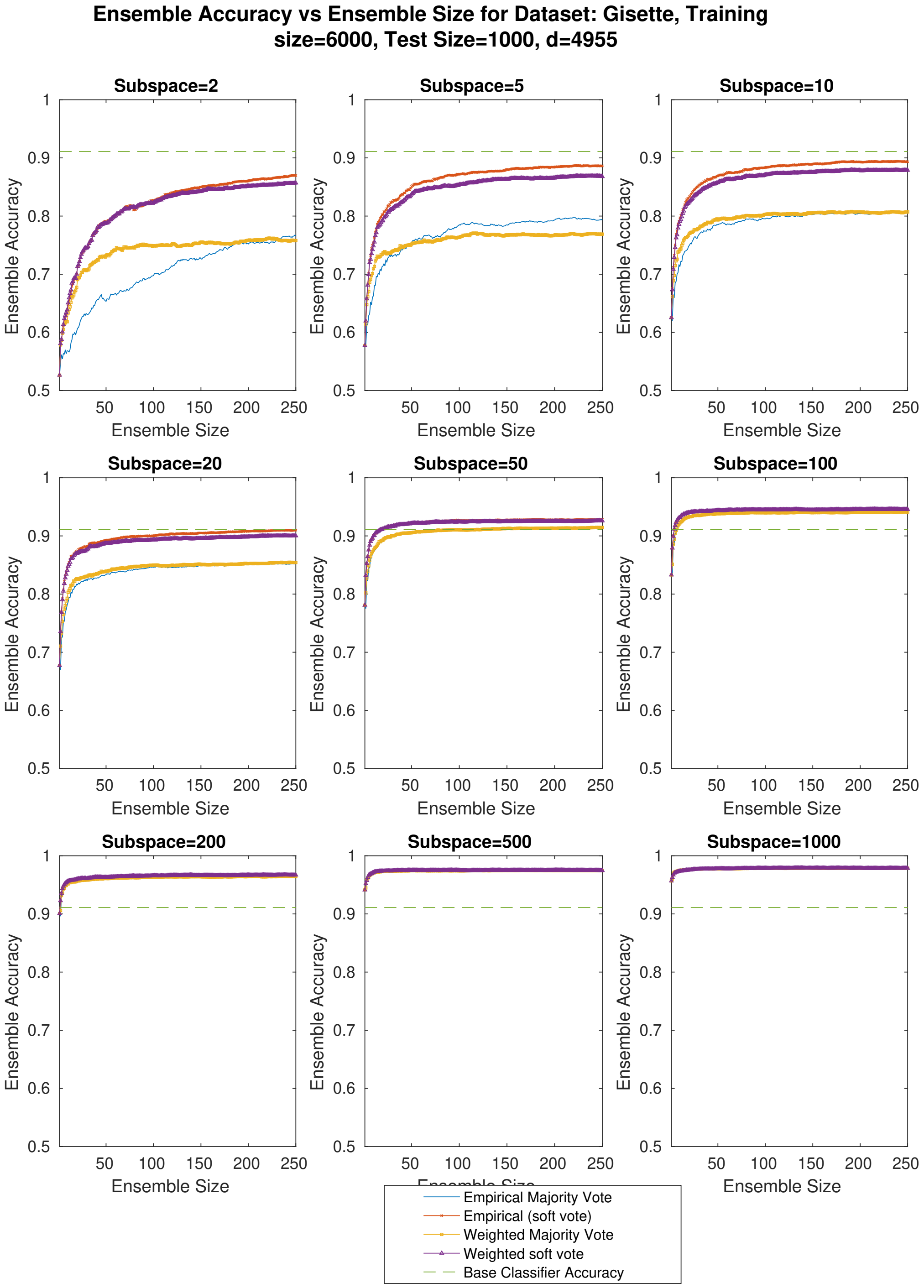

**Figure 6.26:** *Classification accuracy vs ensemble member size for GISETTE. Observe that the RS classification ensemble does well on GISETTE.*

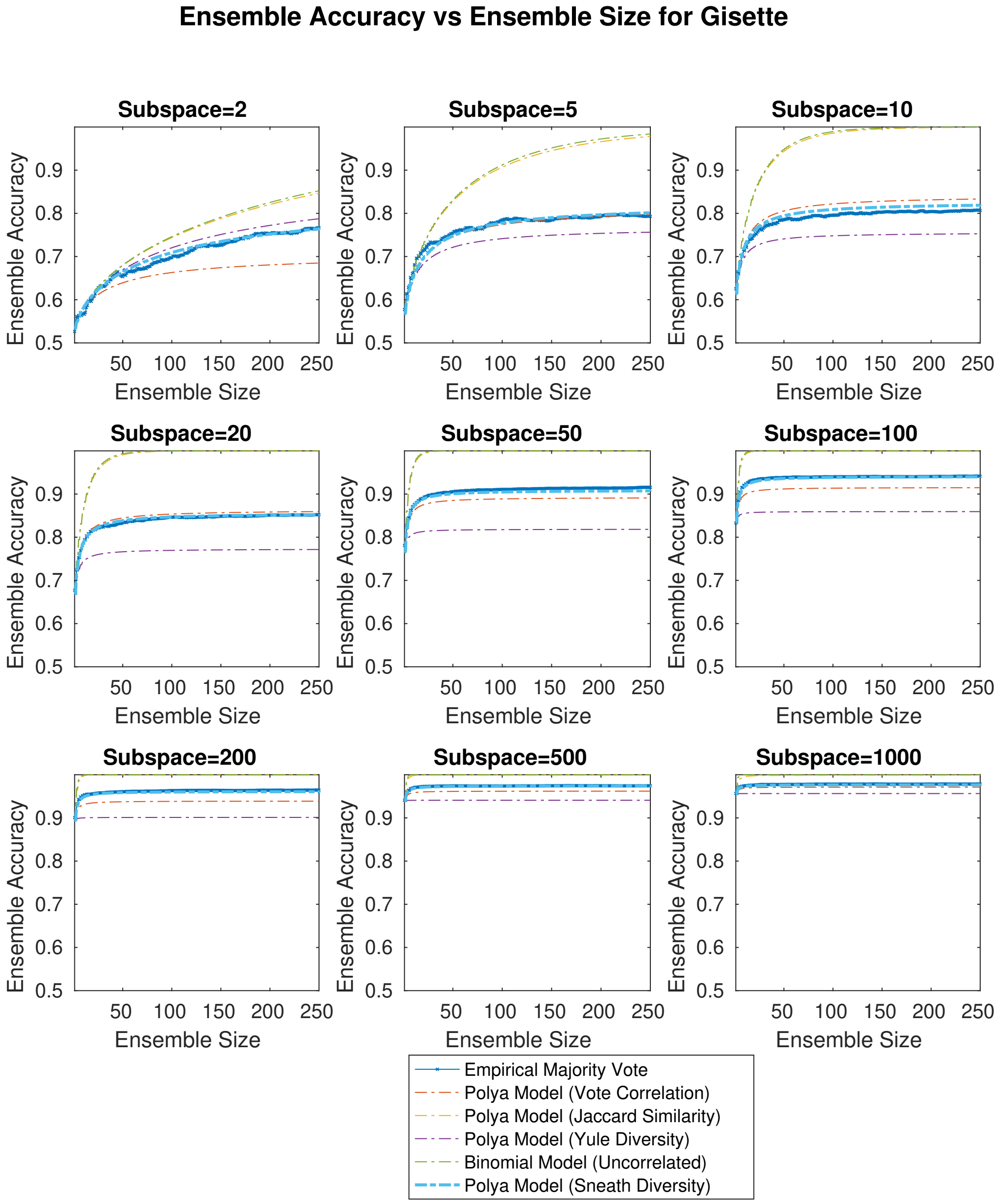

**Figure 6.27:** *Majority vote ensemble accuracy as modelled by a Polya-Eggenberger distribution vs ensemble member size for GISETTE. Observe that our model accurately estimates the majority vote accuracy of the ensemble.*

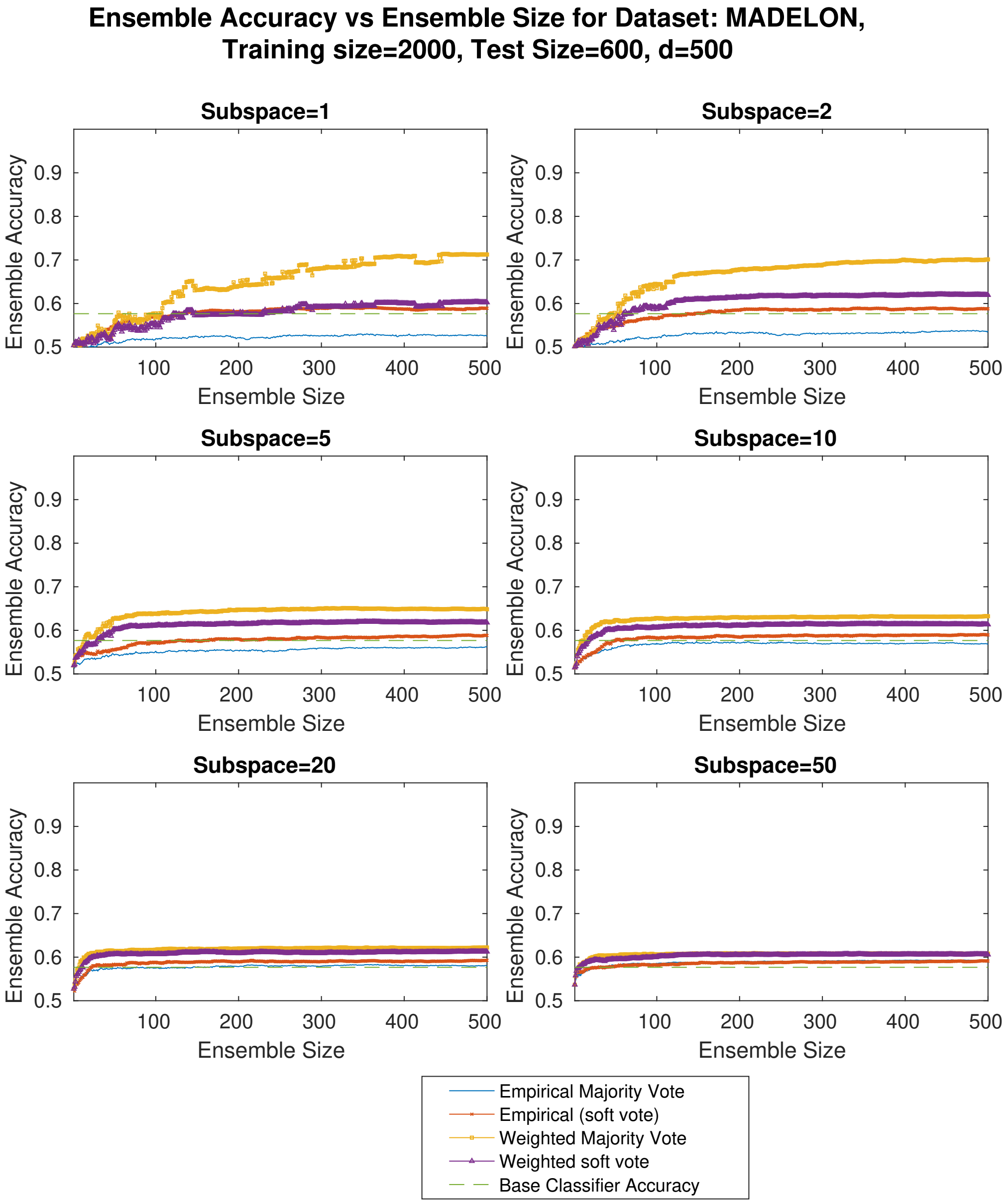

**Figure 6.28:** *Classification accuracy vs ensemble member size for MADELON. Observe that the weighted majority vote on a low number of projection dimension produces the most accurate ensemble classifier among the other choices of classifiers.*

# 7

# Random Subspace as Diversity Generator for Image Classification

**Summary** In this chapter, we apply our theory on Random Subspace Ensembles to image classification tasks, specifically on the Imagenet Large Scale Visual Recognition Challenge dataset.

Taking inspiration from nature, we propose "PseudoSaccades" and show how an ensemble of deep neural network classifiers with PseudoSaccades can give better image classification accuracy compared to a single view classification.

We will also demonstrate how PseudoSaccades can be used to address adversarial examples in deep neural networks, where small imperceivable perturbations can result in incorrect high confidence labels on the classification of the image.

## 7.1 Introduction

Deep Neural Networks (DNN) are state-of-the-art tools for various machine learning tasks (LeCun et al., 2015) and have proved especially useful for image classification tasks. For example, the most recent winners of the Imagenet challenge have all been DNNs.

Although it is not at all well understood — at least in terms of formal learning-theoretic guarantees — how and why DNNs perform so well [1], empirical understanding

[1] For example, while it is known that the VC dimension of a DNN is upper-bounded by the number of nodes in the network (Anthony and Bartlett, 2009), it is not generally known

of how to construct a DNN is substantial and growing, and there are many plausible hypotheses regarding their performance. One striking example of the latter is that not only do DNNs have clear parallels with aspects of human visual processing, but in controlled psychological experiments they also match human performance on visual recognition tasks very closely (Serre et al., 2006).

Taking inspiration from nature, in this chapter we show that an approximate analogue for saccades in human visual processing can improve the performance of a carefully-tuned DNN on an image classification task *that it was explicitly designed to solve*. More precisely, we use a very simple ensemble approach that employs voting but, unlike typical ensemble approaches, rather than learning several similar DNNs and obtaining a weighted combination of votes from that ensemble, instead we use just a *single* DNN but feed it as input multiple random low-dimensional sketches of an image and take the DNN's vote *with itself* on these sketches to reach a majority verdict.

Our approach is inspired by considering saccades in human visual processing, that is eye movements that focus attention on elements in a visual scene. The human eye has only a few degrees of visual arc of high-resolution imaging capability, and saccades are a mechanism by which a scene can be estimated from high-resolution subsampling of parts of it. In human visual processing this subsampling is not uniformly at random – we attend to certain features proportionately more often than others – but we hypothesised that an evolutionary precursor to saccades could have been something closer to a uniform random sampling of features in a scene and that (if indeed there was such a precursor) this must have conferred some selective advantage in order to propagate.

Our results in chapter 4 show that randomly subsampling rows and columns from an image without replacement results in — with high probability — an approximately affine transformation of the original image. Putting these ideas together, since image labels should remain invariant under affine transformations, we speculated that such subsampling could potentially lead to improved classification performance, perhaps even for an already highly-accurate classifier, by providing the classifier with multiple

why DNNs dramatically outperform 'wide' neural networks with the same number of nodes but fewer hidden layers.

low-dimensional sketches of the same image in a similar way that saccadic sampling of a scene does. We call this subsampling of rows and columns 'PseudoSaccades'.

Image classification inputs can be of varying sizes, while most classification algorithms accept only a fixed size input, a common pre-processing is to convert them to a (usually smaller) standard-sized input prior to classification. However as far as we are aware it has not been much exploited before that such pre-processing offers an opportunity for generating multiple instances of a particular image. By extracting PseudoSaccades sketches of an image before applying the standardizing pre-processing, allows the generation – for typical image sizes – of thousands of such instances *per image*. Moreover, unlike cropping and reflection the resulting PseudoSaccades images resemble photographs captured following a change of camera angle and position, while still keeping the subject central in scene – see Figure 7.1-7.4.

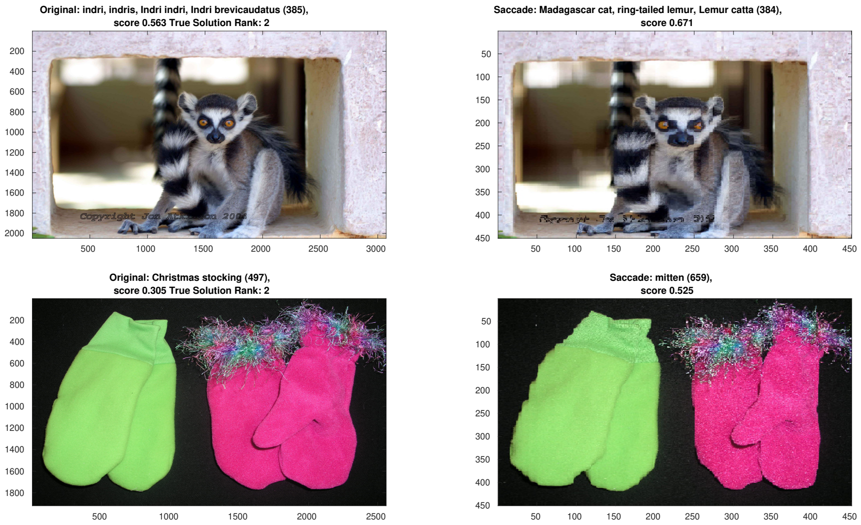

**Figure 7.1:** *Two images incorrectly classified by AlexNet in their original form (left-hand column), but correctly classified in PseudoSaccades form (right-hand column). Observe in Figures 7.1 through 7.3 that the PseudoSaccade view is similar to an image taken from a slightly different camera angle or position.*

Using our simple approach, we obtain statistically significant improvements in classification performance on AlexNet, GoogLeNet, ResNet-50, and ResNet-152 baselines on Imagenet data – e.g. of the order of 0.3% to 0.6% in Top-1 accuracy – essentially nearly for free. We carry out a comprehensive empirical exploration of our approach, reporting results using different levels of subsampling and different ensemble sizes, as well as an initial exploration of whether the improvements have any identifiable systematic component (such as occurring disproportionately in the same class).

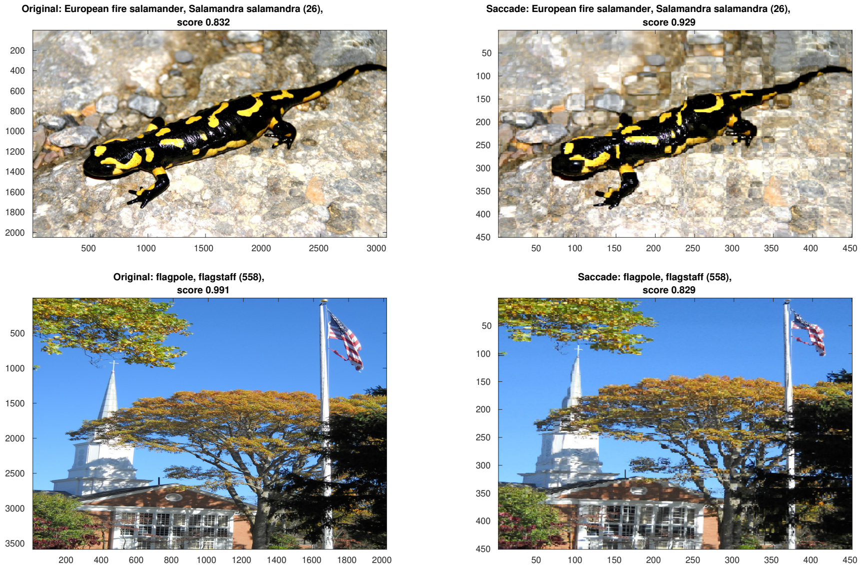

**Figure 7.2:** *Two images correctly classified by AlexNet, both in their original form (left-hand column), and in PseudoSaccades form (right-hand column).*

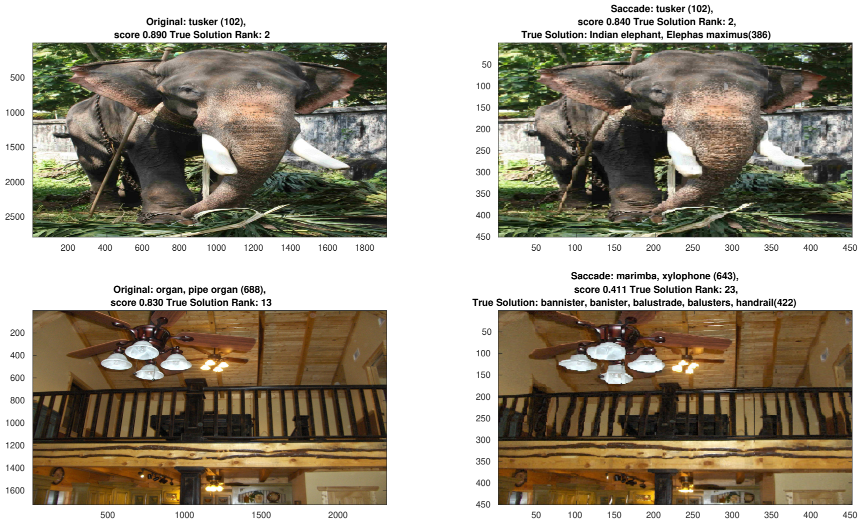
Original: tusker (102),
score 0.890 True Solution Rank: 2
500
1000
1500
2000
2500
200
400
600
800
1000
1200
1400
1600
1800
Saccade: tusker (102),
score 0.840 True Solution Rank: 2,
True Solution: Indian elephant, Elephas maximus(386)
50
100
150
200
250
300
350
400
450
50
100
150
200
250
300
350
400
450
Original: organ, pipe organ (688),
score 0.830 True Solution Rank: 13
200
400
600
800
1000
1200
1400
1600
500
1000
1500
2000
Saccade: marimba, xylophone (643),
score 0.411 True Solution Rank: 23,
True Solution: bannister, banister, balustrade, balusters, handrail(422)
50
100
150
200
250
300
350
400
450
50
100
150
200
250
300
350
400
450

**Figure 7.3:** *Two images incorrectly classified by AlexNet, both in their original form (left-hand column), and in PseudoSaccades form (right-hand column).*

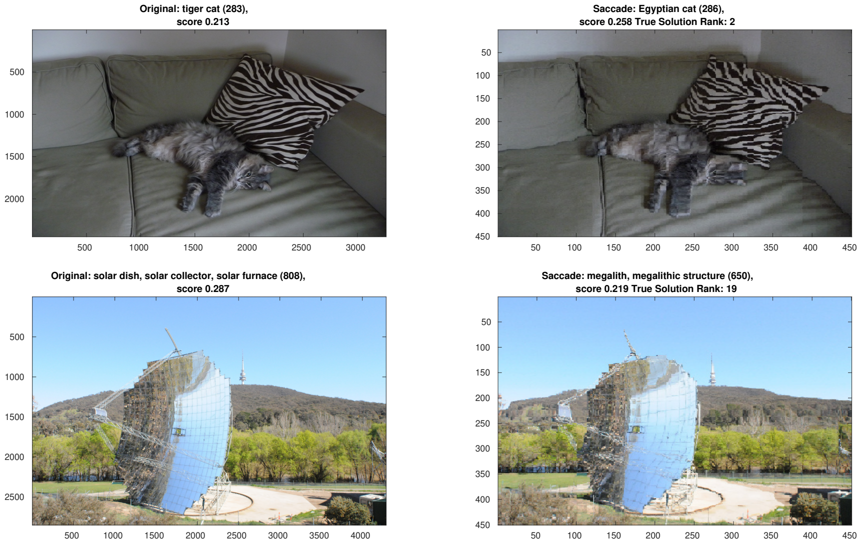

**Figure 7.4:** *Two images correctly classified by AlexNet in their original form (left-hand column), but incorrectly classified in PseudoSaccades form (right-hand column).*

## 7.2 Experiments and Results

In this section, we present details of our experimental protocol and the results of our experiments. We show that the classification accuracy on a single PseudoSaccades version of an image is similar to the accuracy on the original images, given a suitably high projection dimension. Moreover, using PseudoSaccades as a diversity generator, an ensemble classifier employing several PseudoSaccades versions of each image can consistently outperform the classification accuracy of the same classifier on the original images.

### 7.2.1 Dataset and Classifiers

We used the validation dataset from the Imagenet Large Scale Visual Recognition (ILSVR) Challenge 2012 as described in Berg et al. (2010) for our experiments. This dataset comprises of 50000 images, ranging in size from 56x54 pixels to 5005x3646 pixels, where each image is an example from one of 1000 distinct classes. The subject of an image (i.e. the class label) is the dominant and usually central object in that image, and therefore elements of attentive viewing are already present in these images due to the location of the subject. The classes in this dataset range from broad categories to fine-grained labels – for example, one subset of the labels is a classification of 120 different breeds of dogs. Table 7.1 summarizes the main characteristics of this dataset.

We used the winners of ILSVR Challenge from 2012, 2014 and 2015 namely AlexNet (Krizhevsky et al., 2012), GoogLeNet (Szegedy et al., 2015), and ResNet-50 and ResNet-152 (He et al., 2016) to represent the state of art in deep neural network classifiers. These classifiers include many of the latest developments in the evolution of neural networks and each introduced new architectures and other innovations such as ReLU activation functions and skip connections, resulting in the highest accuracies on the Imagenet Large Scale Visual Recognition Challenge for the years 2012, 2014 and 2015 respectively. We used the MATLAB versions of these DNNs implemented in MatConvNet (Vedaldi and Lenc, 2015) and we used the pretrained weights, which are tailored for the ILSVR task to provide a consistent baseline. We note that the pretrained weights for GoogLeNet use weights from Princeton instead of Google, which may affect the accuracy for this DNN compared to the challenge-winning DNN. Also published accuracies in Krizhevsky et al. (2012); Szegedy et al. (2015); He et al. (2016) for the ILSVR challenges are on the challenge test dataset, while we used the validation dataset because it has the labels available. Thus, our central image accuracies for these DNNs show some discrepancies with those published results. Table 7.2 is based on a similar table from Alom et al. (2018) and summarizes the characteristics of these DNNs as well as the baseline accuracies we obtained on the ILSVR challenge validation dataset using them.

### 7.2.2 Experimental Procedure

We classified each image in the ILSVR validation set with no pre-processing, other than that the pre-processing inherent in the DNN itself to standardize the image sizes to obtain baseline accuracies for each of the four DNNs. The pre-processing carried out by the DNNs is noted in table 7.2. We measured the top-1, top-3 and top-5 accuracy for each classifier on the full validation set of 50000 images. These accuracies are also presented in table 7.2 and we will refer to these results obtained on the original images (without subsampling) as the 'baseline classifier' results.

For our PseudoSaccades approach we first fix the 'projection dimension' to be an integer $k \in \{450, 430, 410, 390, 370, 350, 330, 310, 290, 270, 250, 200, 150\}$ and then randomly sample $\min(k, width)$ columns and $\min(k, height)$ rows from the images without replacement. As in the baseline experiments, we apply no further preprocessing, other than that implemented by the DNN to standardize input size, and we measure the top-1, top-3 and top-5 accuracy for each DNN on all 50000 images in the ILSVR validation dataset. We refer to these results as the 'saccade classifier' results. We also store the scores, and the top-5 predicted labels for each combination of sampled projection dimension $k$, image, and DNN. Since the obtained accuracies, scores, and labels are realizations of random variables we repeated these experiments for each combination of $k$, image, and DNN a total of twenty-four times, and we calculated the means and standard deviations for the top-$m$ (top-1, top-3 and top-5) accuracies.

Keeping $k$ fixed we construct an ensemble of size $N \in \{1, 2, \dots, 15\}$ using the scores of between one and fifteen saccade classifiers by sampling without replacement $N$ sets of top-5 scores from the 24 sets of stored saccade classifier scores. We combine these to obtain the ensemble decision by simply summing scores for each label. For each $k, N, m$ triple and each classifier we repeated this process fifty times, and we calculated the corresponding means and standard deviations for the top-1, top-3, and top-5 accuracy.

| | Min | Mean | Max |
|---|---|---|---|
| Image Count | 50000 | | |
| Label Count | 1000 | | |
| Fine-Grained Labels | 120 | | |
| Height | 56 | 430.25 | 5005 |
| Width | 54 | 490.37 | 4288 |
| Size | 3456 | 231320 | 18248230 |

**Table 7.1:** *Summary of the properties of the Imagenet validation dataset.*

### 7.2.3 Results

Table 7.2 gives us the baseline results for the four DNNs. The results for our PseudoSaccades classification ensembles are given in tables 7.3 and 7.4 for ensembles of size 5 and 10 respectively and as well they are plotted in figure 7.5 for all values of $k, N$ and $m$. In figure 7.5 the orange plane shows the baseline accuracy for each

| | AlexNet | GoogLeNet | ResNet-50 | ResNet-152 |
|---|---|---|---|---|
| Architecture | CNN | LeNet | Residual Neural Network | |
| # Convolution Layers | 5 | 57 | 50 | 152 |
| # Fully Connected Layers | 3 | 7 | 1 | 1 |
| # Parameters | 61 M | 7M | 25.6M | 60.3M |
| # Multiply and Accumulates | 724M | 1.43G | 3.9G | 11.3G |
| Regularization | Batch Normalization | Local Response Normalization | Batch Normalization | |
| Image Resizing | bicubic scaling (227x227) | bilinear scaling (224x224) | | |
| Top-1 accuracy | 54.70% | 65.46% | 70.39% | 72.45% |
| Top-3 accuracy | 71.68% | 82.22% | 85.55% | 87.05% |
| Top-5 accuracy | 77.56% | 86.93% | 89.66% | 90.66% |

**Table 7.2:** *Summary of the DNN classifiers.*

classifier and top-$m$ combination within a sub-figure. The surface plots show the average classification error for a given $k, N, m$ triple using PseudoSaccades. From tables 7.3 and 7.4 we see that these average outcomes are very stable indeed, and if the projection dimension $k$ is sufficiently high then even a small ensemble can outperform the DNNs working with the original images at the 5% level of significance (or better) on Top-1, Top-3 and Top-5 classification accuracy. On the other hand, we see that by using a single PseudoSaccades representation of each of the images, we can match or nearly match the baseline accuracy with a projection dimension as high as $k = 350$ (see Figure 7.5). However, with a lower projection dimension, we obtain far worse accuracy than the baseline. The curve comprising the left-hand boundary of each surface plot shows the average accuracy for a single PseudoSaccades plotted against the projection dimension $k$. Finally, we see that the accuracy of the ensemble exceeds that of the baseline classifiers, even for a small ensemble of classifiers and small projection dimension, and this behaviour is consistent across all of the classifier architectures.

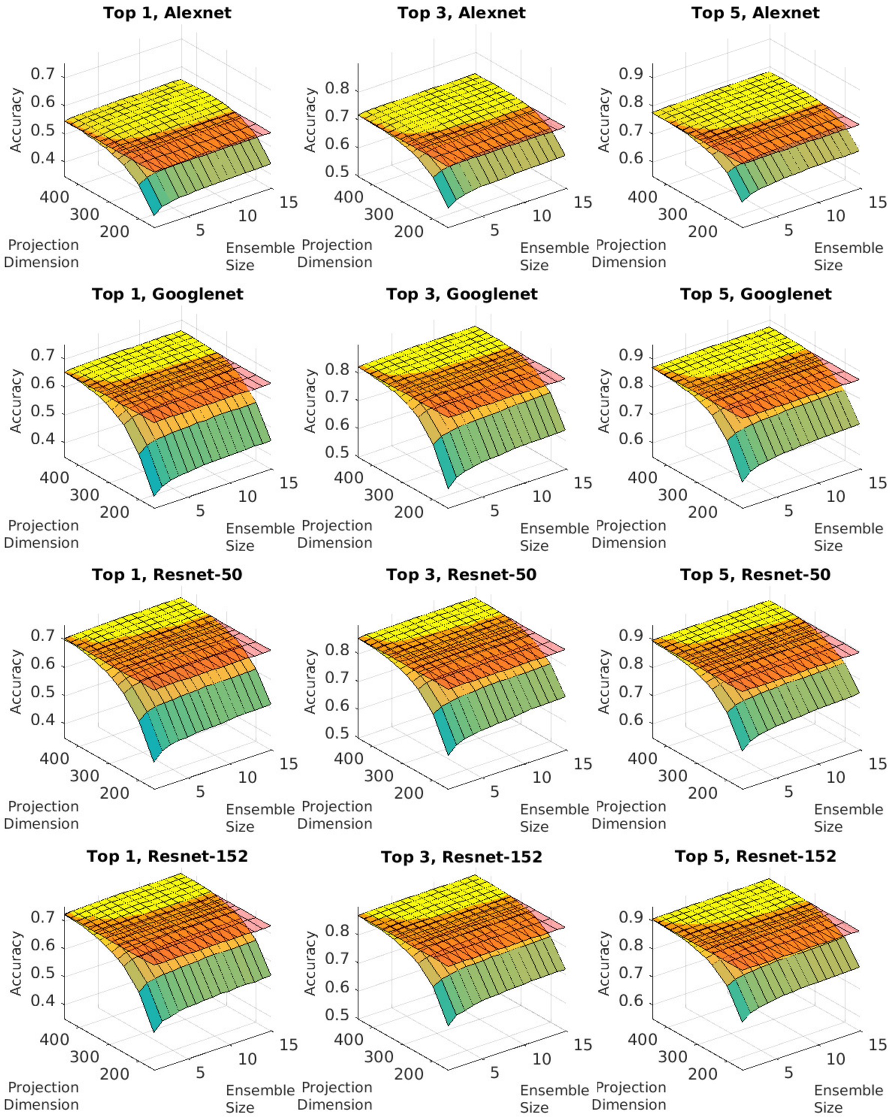

**Figure 7.5:** *Accuracy vs ensemble size and projection dimension. Reference plane shows the accuracy for the baseline classifier. Observe that the accuracy decreases very quickly beyond when $k < k_{min}$, where $k_{min}$ is dependent on the base DNN.*

| | | AlexNet (%) | | GoogLeNet (%) | | ResNet-50 (%) | | ResNet-152 (%) | |
|---|---|---|---|---|---|---|---|---|---|
| Projection Dimensions | | Mean | SD | Mean | SD | Mean | SD | Mean | SD |
| 450 | Top-1 | 55.255 | 0.044 * | 65.918 | 0.051 * | 70.726 | 0.048 * | 72.826 | 0.043 * |
| | Top-3 | 72.235 | 0.033 * | 82.526 | 0.040 * | 85.743 | 0.037 * | 87.289 | 0.050 * |
| | Top-5 | 77.996 | 0.038 * | 87.207 | 0.038 * | 89.830 | 0.049 * | 90.901 | 0.032 * |
| 410 | Top-1 | 55.416 | 0.044 * | 65.910 | 0.047 * | 70.629 | 0.075 * | 72.759 | 0.065 * |
| | Top-3 | 72.342 | 0.041 * | 82.490 | 0.040 * | 85.730 | 0.051 * | 87.264 | 0.043 * |
| | Top-5 | 78.092 | 0.038 * | 87.194 | 0.036 * | 89.771 | 0.057 | 90.860 | 0.038 * |
| 350 | Top-1 | 55.520 | 0.051 * | 65.512 | 0.060 | 69.901 | 0.074 | 72.308 | 0.047 |
| | Top-3 | 72.294 | 0.056 * | 82.153 | 0.050 | 85.250 | 0.069 | 86.955 | 0.043 |
| | Top-5 | 78.061 | 0.045 * | 86.918 | 0.055 | 89.307 | 0.067 | 90.613 | 0.044 |
| 310 | Top-1 | 55.038 | 0.074 * | 64.596 | 0.068 | 68.793 | 0.103 | 71.269 | 0.064 |
| | Top-3 | 71.989 | 0.052 * | 81.429 | 0.047 | 84.286 | 0.080 | 86.253 | 0.047 |
| | Top-5 | 77.738 | 0.057 * | 86.261 | 0.051 | 88.512 | 0.069 | 90.072 | 0.037 |
| 250 | Top-1 | 53.299 | 0.076 | 61.188 | 0.071 | 65.511 | 0.158 | 68.588 | 0.171 |
| | Top-3 | 70.487 | 0.046 | 78.555 | 0.063 | 81.596 | 0.102 | 84.113 | 0.112 |
| | Top-5 | 76.398 | 0.066 | 83.878 | 0.057 | 86.247 | 0.068 | 88.283 | 0.076 |

**Table 7.3:** *Ensemble classifier accuracy for ensemble size $N = 5$ and projection dimensions $k \in \{450, 410, 350, 310, 250\}$, with the standard deviation from a sample of 50 ensembles. Values with '*' exceeded the top-k accuracies of the baseline classifiers by at least 2 standard deviations. The baseline accuracy for each of the DNNs are as noted in Table 7.2.*

| | | AlexNet (%) | | GoogLeNet (%) | | ResNet-50 (%) | | ResNet-152 (%) | |
|---|---|---|---|---|---|---|---|---|---|
| Projection Dimensions | | Mean | SD | Mean | SD | Mean | SD | Mean | SD |
| 450 | Top-1 | 55.348 | 0.035 * | 65.987 | 0.038 * | 70.828 | 0.032 * | 72.894 | 0.026 * |
| | Top-3 | 72.307 | 0.032 * | 82.582 | 0.036 * | 85.802 | 0.028 * | 87.365 | 0.029 * |
| | Top-5 | 78.081 | 0.030 * | 87.266 | 0.031 * | 89.903 | 0.035 * | 90.953 | 0.024 * |
| 410 | Top-1 | 55.568 | 0.048 * | 66.003 | 0.041 * | 70.734 | 0.055 * | 72.863 | 0.043 * |
| | Top-3 | 72.435 | 0.031 * | 82.568 | 0.034 * | 85.828 | 0.026 * | 87.361 | 0.028 * |
| | Top-5 | 78.195 | 0.035 * | 87.278 | 0.033 * | 89.871 | 0.039 * | 90.938 | 0.025 * |
| 350 | Top-1 | 55.716 | 0.048 * | 65.672 | 0.041 * | 70.065 | 0.047 | 72.500 | 0.040 |
| | Top-3 | 72.444 | 0.035 * | 82.310 | 0.036 | 85.385 | 0.048 | 87.118 | 0.025 |
| | Top-5 | 78.233 | 0.036 * | 87.069 | 0.033 * | 89.460 | 0.046 | 90.747 | 0.025 * |
| 310 | Top-1 | 55.300 | 0.068 * | 64.866 | 0.043 | 69.038 | 0.054 | 71.497 | 0.043 |
| | Top-3 | 72.203 | 0.055 * | 81.660 | 0.045 | 84.524 | 0.073 | 86.457 | 0.036 |
| | Top-5 | 77.966 | 0.045 * | 86.494 | 0.044 | 88.733 | 0.054 | 90.304 | 0.025 |
| 250 | Top-1 | 53.643 | 0.050 | 61.604 | 0.055 | 65.864 | 0.121 | 68.956 | 0.093 |
| | Top-3 | 70.803 | 0.046 | 78.955 | 0.056 | 81.955 | 0.073 | 84.484 | 0.082 |
| | Top-5 | 76.728 | 0.046 | 84.268 | 0.048 | 86.639 | 0.071 | 88.624 | 0.066 |

**Table 7.4:** *The mean ensemble classifier accuracy for ensemble size $N = 10$ for projection dimensions $k \in \{450, 410, 350, 310, 250\}$, with the standard deviation from a sample of 50 ensembles. Values with '*' exceeded the top-m accuracies of the baseline classifiers by at least 2 standard deviations. The baseline accuracy for each of the DNNs are as noted in Table 7.2.*

### 7.2.4 Further experiments

A natural question, given the improvements from PseudoSaccades, is whether an 'ensemble of ensembles' would improve performance further? We started by looking further into the diversity of the saccade classifiers. In line with our results in chapter 6, we used the Sneath and Sokal (1963) diversity measure to calculate the correlation between the saccade classifier errors and the baseline classifier errors using

$$\rho_{i,j} = \frac{N_{11}N_{00} - N_{01}N_{10}}{\sqrt{(N_{11}+N_{10})(N_{01}+N_{00})(N_{11}+N_{01})(N_{10}+N_{00})}}$$

, where $i$ is the base classifier, and $j$ is the saccade classifier. In table 7.5, we see that – based on this summary statistic – the accuracy of the saccade classifiers is highly correlated with that of the corresponding baseline classifier, indicating to us that the classifier performance is not substantially reduced by PseudoSaccades projection. Table 7.6 meanwhile shows that although the saccade classifier errors are correlated with one another, this is to a lesser degree than to the baseline classifiers. These facts suggest that there might be little to gain from combining the PseudoSaccades ensembles from different DNNs into a larger ensemble. However, since all of the accuracies are already high it seemed worthwhile to examine where the improvements were coming from - were these for similar class labels for every classifier for example?

Digging deeper we observed that the classification accuracy of the individual classes is not uniformly affected by PseudoSaccades. Moreover, at this lower level of granularity, we see that the different architectures do tend to be affected by the PseudoSaccades differently.

Tables 7.10,7.11 and 7.9 show lists of predicted class labels for a given class label for ResNet-152, with projection dimension 390 and ensemble size 5. Note that there are 50 instances in each of the true class labels, and we omitted predicted labels where there was only a single prediction or two predictions for reasons of space and readability.

In table 7.10, we present a list of labels for which the ResNet-152 classifier obtained less than 20% recall. We obtained similar tables for the other three classifiers which we deferred to the appendix (Tables. D.8 - D.16). We observed that the ensemble of PseudoSaccades classifiers performs similarly to the baseline classifier on labels that are also difficult for the baseline classifier to predict accurately, but we also saw that the different classifier architecture has their own sets of 'difficult labels' that are different.

Finally, tables 7.9 and 7.11 give examples of class labels where the ensemble classifier respectively gives either a large improvement or is much worse (±10%) on classification accuracy for these classes. We found that the saccade classifiers were affected differently on different classifier architecture.

Thus, although high-level summary statistics seemed to indicate little diversity between the different ensemble classifiers, a more principled investigation reveals that the errors for both the original DNNs and the corresponding PseudoSaccades ensembles arise from different classes and different instances in the dataset.

We, therefore, constructed two ensemble classifiers - one using the four baseline DNNs and one that combined four PseudoSaccades ensembles seeing if further improvements were possible. We used five-fold cross-validation on the validation set data to train a shallow neural network with a single hidden layer with ReLU activations on the baseline scores for 40000 images from the validation dataset to learn a weighting function for the ensemble of baseline classifiers. We used the average and maximum scores from PseudoSaccades versions of the four DNNs for the same 40000 images to train a similar network to weight the 'ensemble of ensembles'. We evaluated both ensembles using the 10000 remaining held-out images from the validation dataset and estimated the top-1,top-3 and top-5 accuracies for both ensembles with the cross-validation error. We carried out one round of five-fold cross-validation for the baseline classifiers and 50 rounds for the PseudoSaccades classifiers, for different $k, N, m$ triples and calculated the mean accuracies and their standard deviations. For both sets of ensembles, we saw substantial improvements over the original baseline accuracies and, consistent with our earlier experiments, the PseudoSaccades ensembles were yet again able to outperform the ensemble of baseline DNN classifiers. Figure 7.5 shows the accuracy of the DNN ensemble versus the PseudoSaccades ensembles for different $k, N, m$ triples. The horizontal orange plane indicates the (average) accuracy of the DNN ensemble. The PseudoSaccades ensembles outperform the increased Top-1 accuracy baseline of 75.78% by 0.3%, and the accuracy of the best performing classifier ResNet-152 by 3.7%. We conjecture that further, possibly minor, improvements in accuracy may be possible using a more careful approach to learn the weighting function.

| Saccade Dimensions | AlexNet | GoogLeNet | ResNet 50 | ResNet 152 |
|---|---|---|---|---|
| 450 | 0.8959 | 0.8952 | 0.8916 | 0.8949 |
| 430 | 0.8851 | 0.882 | 0.877 | 0.8811 |
| 410 | 0.8753 | 0.8694 | 0.8615 | 0.8681 |
| 390 | 0.865 | 0.8552 | 0.8479 | 0.8552 |
| 370 | 0.8524 | 0.8368 | 0.8301 | 0.8377 |
| 350 | 0.8375 | 0.8164 | 0.8078 | 0.8164 |
| 330 | 0.8221 | 0.7937 | 0.7853 | 0.7943 |
| 310 | 0.8039 | 0.7683 | 0.7601 | 0.7715 |
| 290 | 0.785 | 0.7418 | 0.7335 | 0.7449 |
| 270 | 0.764 | 0.7119 | 0.7068 | 0.7174 |
| 250 | 0.742 | 0.678 | 0.676 | 0.6889 |
| 200 | 0.6727 | 0.5729 | 0.5806 | 0.5977 |
| 150 | 0.5626 | 0.437 | 0.4563 | 0.4666 |

**Table 7.5:** *Average classifier correlation $\rho_{base,saccade}$ between baseline classifier and saccade classifiers.*

## 7.3 Comparison to existing methods

While PseudoSaccades performs marginally worse than the 10-crop method used in ResNet-50 (He et al., 2016) on average, PseudoSaccades outperforms the multi-crop approach used in GoogLeNet (Szegedy et al., 2015). Table 7.8 summarises the accuracy of the PseudoSaccades ensemble versus a 10-crop ensemble. If a single image rather than the 10 cropped view were not used for classification for GoogLeNet or ResNet, the performance drops sharply. We conjectured that PseudoSaccades could be more robust than the 10-crop approach and in section 7.4 we explore this intuition for adversarial examples.

We conjectured that the difference in the performance gains come from the ensemble diversity. Table 7.7 summarises the ensemble diversity for GoogLeNet and ResNet-50 using the correlation measure $\rho$ described in section 7.2.4. A saccade

| Saccade Dimensions | AlexNet | GoogLeNet | ResNet 50 | ResNet 152 |
|---|---|---|---|---|
| 450 | 0.8847 | 0.8830 | 0.8809 | 0.8876 |
| 430 | 0.8744 | 0.8702 | 0.8682 | 0.8754 |
| 410 | 0.8657 | 0.8597 | 0.8562 | 0.8648 |
| 390 | 0.8564 | 0.8484 | 0.8456 | 0.8543 |
| 370 | 0.8448 | 0.8344 | 0.8337 | 0.8419 |
| 350 | 0.8320 | 0.8202 | 0.8196 | 0.8266 |
| 330 | 0.8196 | 0.8060 | 0.8059 | 0.8121 |
| 310 | 0.8077 | 0.7917 | 0.7938 | 0.7991 |
| 290 | 0.7956 | 0.7771 | 0.7792 | 0.7849 |
| 270 | 0.7835 | 0.7621 | 0.7667 | 0.7700 |
| 250 | 0.7717 | 0.7466 | 0.7528 | 0.7572 |
| 200 | 0.7369 | 0.7027 | 0.7138 | 0.7211 |
| 150 | 0.6942 | 0.6473 | 0.6666 | 0.6750 |

**Table 7.6:** *Average classifier correlation $\rho_{saccade_1,saccade_2}$ between all pairs of saccade classifiers.*

| | average$(\rho_i, \rho_j)$ |
|---|---|
| ResNet 50 | 0.7635 |
| GoogLeNet | 0.6920 |
| PseudoSaccades (450 Saccade Dimensions) | 0.8840 |

**Table 7.7:** *Average Classifier Correlation of 450 Saccade Dimension PseudoSaccades versus 10-crop view. Observe that the diversity measure $\rho$ for PseudoSaccades is larger than the diversity measure for the 10-crop DNNs.*

|  | Top-1 | Top-3 | Top-5 |
|---|---|---|---|
| 10 Crop GoogLeNet | 67.81 | 84.11 | 88.58 |
| 10 Crop ResNet-50 | 76.68 | 89.95 | 93.17 |
| 10 Ensembles of<br>PseudoSaccades<br>450 Saccade Dimensions | 76.16 | 88.13 | 90.86 |

**Table 7.8:** *Ensemble accuracy of PseudoSaccades versus 10-crop view image classification. Observe that PseudoSaccades outperforms 10 crop GoogLeNet, while performing marginally worse than 10-crop ResNet-50*

dimension of 270, would have given approximately the same diversity measure as the 10-crop view. However, lowering the saccade dimension to 270 incurs too large of an accuracy loss ($>$ 10%) for the ensemble to recover. We also observe that the DNNs were trained using the 10-crop view, while we were using the pretrained network as it is without retraining. We conjecture that if the DNN were to be retrained with PseudoSaccades views of the training examples, this could provide for a low-cost approach for data augmentation that would be reflected in this gap closing. However, this remain for future work.

## 7.4 Adversarial Examples

A particularly interesting weakness of deep neural networks is their susceptibility to adversarial examples (Szegedy et al., 2013). Adversarial examples are non-random perturbations added to input specifically in order to maximize the prediction error.

In the context of image recognition tasks, an adversarial example could result in the classifiers giving very high confidence prediction of an incorrect label to an image that is usually imperceptibly different from the "clean" image.

Recent literatures shows that there has been significant interest in adversarial examples especially for important common tasks such as speech-to-text and image recognition tasks (i.e. Carlini and Wagner (2017, 2018); Gu and Rigazio (2014); Athalye et al. (2017); Goodfellow et al. (2014b); Kurakin et al. (2016) ) This interest in adversarial examples may be due to the inherent dangers of high confidence misclassification, for instance, misidentifying civilian-targets as military-targets or a

| **true label** | **base label** | **saccade label** | **ensemble label** |
|---|---|---|---|
| **rock crab** | **rock crab (28)**<br>dungeness crab (4)<br>hermit crab (4) | **rock crab (30)**<br>crayfish (3)<br>hermit crab (3) | **rock crab (33)**<br>dungeness crab (3)<br>hermit crab (4) |
| **bedlington terrier** | **bedlington terrier (28)** | **bedlington terrier (42)** | **bedlington terrier (43)** |
| **labrador retriever** | **labrador retriever (34)**<br>bloodhound (3)<br>saluki (3)<br>golden retriever (3) | **labrador retriever (36)**<br>saluki (3) | **labrador retriever (39)** |
| **bell cote** | **bell cote (26)**<br>chime (3)<br>church (11)<br>monastry (4) | **bell cote (31)**<br>church (8)<br>monastry (4) | **bell cote (31)**<br>church (9)<br>monastry (4) |
| **bow** | **bow (30)** | **bow (32)** | **bow (35)** |
| **necklace** | **necklace (40)**<br>chain (3) | **necklace (46)** | **necklace (46)** |
| **pitcher** | **pitcher (19)**<br>vase (4)<br>water jug (6) | **pitcher (25)**<br>vase(4)<br>water jug (5) | **pitcher (24)**<br>vase (3)<br>water jug (8) |
| **plastic bag** | **plastic bag (24)** | **plastic bag (26)** | **plastic bag (29)** |
| **hen of the wood** | **hen of the wood (35)**<br>coral fungus (3) | **hen of the wood (36)**<br>coral fungus (4) | **hen of the wood (40)** |

**Table 7.9:** *Labels where ensemble method performed significantly better* ($\geq$ 10%) *than the baseline ResNet-152 Imagenet classifier. Number of instances for which the given label was returned by classifier in brackets. Note that there are 50 instances for each of the classes. We omitted all predictions that occurred two or less times therefore the sum of the instances does not total to 50.*

| **true label** | **base label** | **saccade label** | **ensemble label** |
|---|---|---|---|
| **cassette player** | **cassette player (10)**<br>cd player (4)<br>radio (3)<br>tape player (22) | **cassette player (9)**<br>cd player (4)<br>tape player (23) | **cassette player (9)**<br>cd player (4)<br>radio (3)<br>tape player (21) |
| **crt screen** | **crt screen (8)**<br>desk (6)<br>desktop computer (8)<br>monitor (4)<br>television (8) | **crt screen (9)**<br>desk (6)<br>desktop computer (8)<br>monitor (5)<br>television (9) | **crt screen (9)**<br>desk (5)<br>desktop computer (8)<br>laptop computer (3)<br>monitor (6)<br>television (9) |
| **sunglass** | **sunglass (11)**<br>sunglasses (19) | **sunglass (10)**<br>sunglasses (19) | **sunglass (11)**<br>sunglasses (16) |

**Table 7.10:** *Labels where ResNet-152 Imagenet classifier achieved* $\leq 20\%$ *recall. Number of instances for which the given label was returned by classifier in brackets. Note that there are 50 instances for each of the classes. We omitted all predictions that occurred two or less times therefore the sum of the instances does not total to 50.*

| **true label** | **base label** | **saccade label** | **ensemble label** |
|---|---|---|---|
| **mantis** | **mantis (37)**<br>walking stick (3) | **mantis (31)**<br>walking stick (5) | **mantis (32)**<br>walking stick (5) |
| **abaya** | **abaya (41)** | **abaya (37)**<br>cloak (3) | **abaya (36)**<br>cloak (3) |
| **perfume** | **perfume (40)** | **perfume (35)** | **perfume (34)** |
| **wok** | **wok (28)**<br>hot pot (10) | **wok (22)**<br>dutch oven (4)<br>frying pan (3)<br>hot pot (9) | **wok (23)**<br>frying pan (4)<br>hot pot (11) |

**Table 7.11:** *Labels where ensemble method performed significantly worse* $(\geq 10\%)$ *than the baseline ResNet-152 Imagenet classifier. Number of instances for which the given label was returned by classifier in brackets. Note that there are 50 instances for each of the classes. We omitted all predictions that occurred two or less times therefore the sum of the instances does not total to 50.*

failure to correctly identify traffic signs in self-driving automobiles (Sitawarin et al., 2018). The fact that it is not difficult for bad-actors (or bad luck) to cause failure in DNNs to classify correctly may also be reasons for interest in research in adversarial examples.

Inspired by our results in chapter 6, in particular the empirical results demonstrating that RS ensembles improve tolerance to mislabelled examples, we explore the robustness of PseudoSaccades to adversarial examples using ResNet-50. Our empirical findings show that PseudoSaccades improve robustness to adversarial examples. Moreover, this improvement in robustness does not require retraining of the neural network and can be used with existing pretrained DNN weights. Finally, this robustness persists under a wide range of adversarial attacks.

### 7.4.1 Experimental setup and result

We used Foolbox (Rauber et al., 2017), a Python-based toolkit to generate adversarial examples for DNNs. We set the Foolbox model to use ResNet-50 as the base DNN, and used three images, namely of a giant panda, a hen and a jay. ResNet-50 was chosen as it is sufficiently accurate as a baseline. It is far quicker to generate adversarial examples for ResNet-50 than for ResNet-152, and we expect the outcomes to be similar. [2]

We categorized Foolbox attacks to three categories of attacks based on the perceptible changes caused by the attack. The first category of the attacks are we dubbed “gradient-based attacks”. The attacks in this category use the gradients of the pretrained neural network to add perturbations that maximize the loss with respect to the image. Gradient-based attacks are usually difficult to perceive visually, and some gradient-based attacks allow for a targeted attack where the attacker can choose the target class for the adversarial example. The second category of attacks is what we call “pixel-based attacks“. Pixel-based attacks add high contrast pixels to the image until the classifier fails to classify the image correctly. An example of this attack is “Salt-and-Pepper” attack, where black or white pixels are added at random until the image is misclassified.

[2] Generating the full adversarial examples for ResNet-50 took approximately 45 hours

The last category of attacks is what we call “contrast-based” attacks. In this category of attacks, the image is visually degraded — for example by adding Gaussian noise, decreasing the contrast, or blurring the image — until the classifier misclassified the image. Note that we used a different definition for the categories of attacks than the documentation for Foolbox — the Foolbox documentation grouped both “pixel-based-attacks” and “contrast-based” attacks as “Decision-based attacks”. Table 7.12 summarizes the types of attacks used as well as the settings used for the attack. The first twenty attacks are “gradient-based attacks”, the next two are “pixel-based attacks” and the last five are “contrast-based” attacks.

We then generated saccade views of 200 × 200 pixels and classify the saccade views of the adversarial examples using ResNet-50. We then repeat this process 30 times to ensure our results are consistent. We also classified the adversarial examples without applying PseudoSaccade using ResNet-50 as a baseline. Table 7.13 shows the average true label scores of the saccade views. Figures 7.6, 7.8 and 7.9 shows the results comparison of the classification between the baseline, and the saccade view, with the middle image illustrating the additional noise added to the image to generate the adversarial example. For these images, we scaled up the pixel value in the difference image for visual clarity purposes. Additional figures (Figures D.36-D.41) are available in the appendix for visual comparison.

### 7.4.2 Results and Discussion

PseudoSaccades appears to be very robust against gradient-based adversarial examples. Figures 7.6 show some of the adversarial examples and the labels predicted by ResNet-50 and the labels of the saccade view of the adversarial example. Surprisingly, for the test example “giant panda” using “Gradient” attack, we see that the classification of the saccade view gives the correct label with a lower than expected score. We have no reasonable explanation as to why this is so, although we observed that the adversarial example is visually degraded in comparison to the “clean” image, and we suspect that it may be due to how the adversarial example was generated instead. We note however, that the saccade view was able to recover the correct label, while the baseline classification still gave an incorrect label. See figure 7.7 for a visual comparison of the classification.

PseudoSaccades also appear to be resistant to pixel-based attacks as shown in Figures 7.8. Classifications using the saccade view are able to recover the correct labels, with reasonably high confidence, though not as high as in the case of saccade views of gradient-based attacks

However, PseudoSaccades are not as robust against contrast-based attacks as shown in Figures 7.9. While PseudoSaccades can sometimes recover the correct label more often than the baseline classification, classification using the saccade views on the adversarial example usually result in either low confidence predictions of the correct label or low confidence predictions with an incorrect label. .

Our results are broadly consistent with the findings of Gurbaxani and Mishra (2018) which shows using small perturbations of the image to be classified can defeat adversarial examples. However, they do not consider RS as a remedy, and their analysis does not describe the effect of different potential remedies on the classifier scores or examine their robustness. For example, if one is unsure if an example is adversarial, it is not clear how or if one should use their approaches.

Our findings here also lend support to the conjecture of Elsayed et al. (2018) that saccades in human vision are one reason why humans are not as susceptible to the same type of adversarial examples in deep neural networks. If would be extremely gratifying if further study of PseudoSaccades revealed some additional interesting insights into how our human visual cortex processes information and help to build neural network systems that mimic human vision more faithfully, though such an undertaking is beyond the capabilities of the author.

## 7.5 Conclusions and future work

We demonstrated that using a very simple, and computationally cheap, 'PseudoSaccades' ensemble learning approach could improve the image classification performance of DNNs. This improvement is small but statistically significant at the 5% level and requires no retraining of the neural network. Following a careful analysis of the sources of error in our classification problem, we showed that these improvements also propagate to a weighted ensemble of PseudoSaccades versions of (off-the-shelf) DNNs.

| Attack Type | Criterion |
|---|---|
| Gradient | Top-K ($k \geq 2$) |
| GradientSign | Top-K ($k \geq 2$) |
| IterativeGradient | Top-K ($k \geq 2$) |
| IterativeGradientSign | Top-K ($k \geq 2$) |
| LBFGS | Targeted ($p \geq 0.8$) |
| DeepFool | Top-K ($k \geq 2$) |
| DeepFoolL2 | Top-K ($k \geq 2$) |
| DeepFoolLinfinity | Top-K ($k \geq 2$) |
| SaliencyMap | Targeted ($p \geq 0.8$) |
| CarliniWagnerL2 | Targeted ($p \geq 0.8$) |
| LinfinityBasicIterative | Targeted ($p \geq 0.8$) |
| BasicIterativeMethod | Targeted ($p \geq 0.8$) |
| L1BasicIterative | Targeted ($p \geq 0.8$) |
| L2BasicIterative | Targeted ($p \geq 0.8$) |
| ProjectedGradientDescentAttack | Targeted ($p \geq 0.8$) |
| ProjectedGradientDescent | Targeted ($p \geq 0.8$) |
| RandomStartProjectedGradientDescent | Targeted ($p \geq 0.8$) |
| RandomProjectedGradientDescent | Targeted ($p \geq 0.8$) |
| MomentumIterative | Targeted ($p \geq 0.8$) |
| MomentumIterativeMethod | Targeted ($p \geq 0.8$) |
| SaltAndPepperNoise | Top-K ($k \geq 2$) |
| Pointwise | Top-K ($k \geq 2$) |
| GaussianBlur | Top-K ($k \geq 2$) |
| ContrastReduction | Top-K ($k \geq 2$) |
| AdditiveUniformNoise | Top-K ($k \geq 2$) |
| AdditiveGaussianNoise | Top-K ($k \geq 2$) |
| BlendedUniformNoise | Top-K ($k \geq 2$) |

**Table 7.12:** *List of attack types and criterion used to generate the adversarial examples. The first 20 attacks are "gradient-based" attacks, the next two are "pixel-based" attacks and the last five are "contrast-based" attacks.*

|  | Hen | Giant Panda | Jay |
|---|---|---|---|
| GradientAttack | 0.6211 | 0.182 | 0.5818 |
| GradientSignAttack | 0.7094 | 0.3157 | 0.7015 |
| IterativeGradientAttack | 0.6785 | 0.6691 | 0.6398 |
| IterativeGradientSignAttack | 0.734 | 0.9124 | 0.6742 |
| LBFGSAttack | 0.8964 | 0.9954 | 0.9051 |
| DeepFoolAttack | 0.7966 | 0.8972 | 0.8795 |
| DeepFoolL2Attack | 0.8055 | 0.8652 | 0.8808 |
| DeepFoolLinfinityAttack | 0.814 | 0.8972 | 0.9219 |
| SaliencyMapAttack | 0.7562 | 0.7869 | 0.3547 |
| CarliniWagnerL2Attack | 0.8501 | 0.9262 | 0.659 |
| LinfinityBasicIterativeAttack | 0.8482 | 0.9582 | 0.8462 |
| BasicIterativeMethod | 0.8612 | 0.948 | 0.8415 |
| L1BasicIterativeAttack | 0.8388 | 0.89 | 0.7829 |
| L2BasicIterativeAttack | 0.8517 | 0.8703 | 0.7289 |
| ProjectedGradientDescentAttack | 0.8525 | 0.9645 | 0.8406 |
| ProjectedGradientDescent | 0.8401 | 0.9628 | 0.8463 |
| RandomStartProjectedGradientDescentAttack | 0.8406 | 0.972 | 0.8934 |
| RandomProjectedGradientDescent | 0.8203 | 0.9725 | 0.8448 |
| MomentumIterativeAttack | 0.7907 | 0.9132 | 0.7894 |
| MomentumIterativeMethod | 0.7852 | 0.9118 | 0.8239 |
| SaltAndPepperNoiseAttack | 0.5201 | 0.1078 | 0.5843 |
| PointwiseAttack | 0.7539 | 0.8414 | 0.6981 |
| GaussianBlurAttack | 0.1339 | 0.1111 | 0.0307 |
| ContrastReductionAttack | 0.3267 | 0.2763 | 0.1452 |
| AdditiveUniformNoiseAttack | 0.0959 | 0.0597 | 0.1764 |
| AdditiveGaussianNoiseAttack | 0.1044 | 0.2131 | 0.1039 |
| BlendedUniformNoiseAttack | 0.0128 | 0.0094 | 0.0087 |

**Table 7.13:** *Summary of the true label scores of the saccade views averaged over 30 runs. Note that the classifier returned high confidence scores on the true label scores on for "gradient-based" adversarial example attacks on the PseudoSaccade views, and low scores for the "contrast-based' adversarial example attacks.*

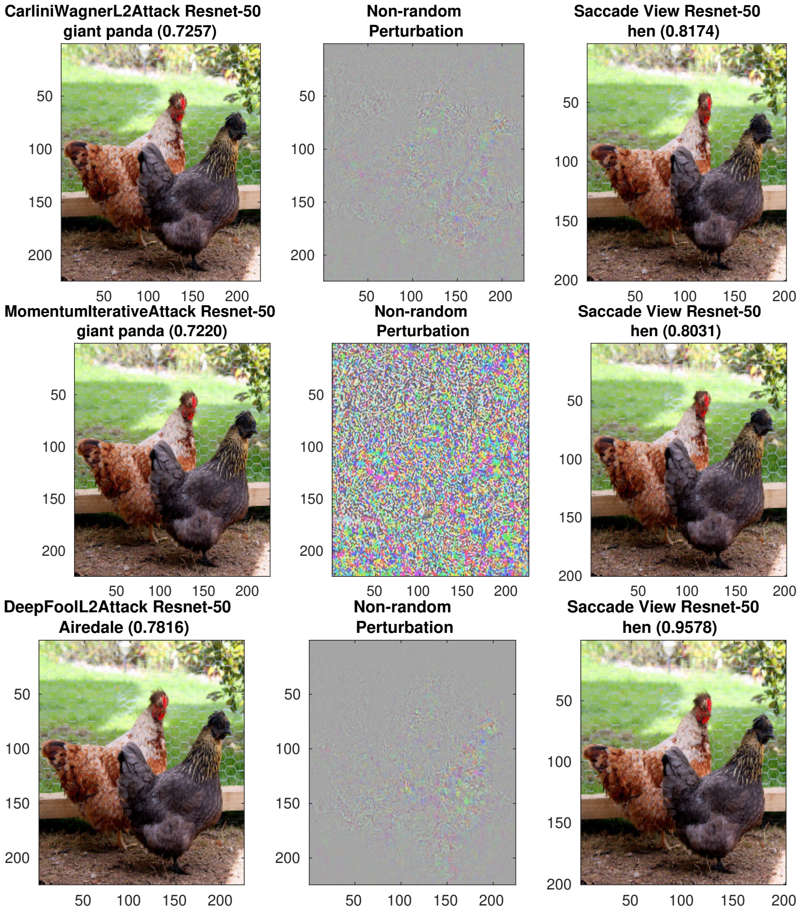

**Figure 7.6:** *ResNet-50 classification of "gradient-based" adversarial attacks on the original form (left), and PseudoSaccade form (right). Image in the centre column is a visual representation of the adversarial attack. Observe that classification is very robust on the PseudoSaccades forms, returning a high confidence prediction on the true label.*

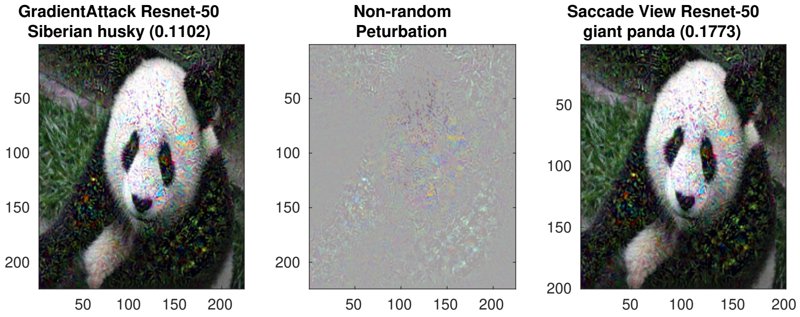

**Figure 7.7:** *ResNet-50 classification of "giant panda" with "GradientAttack" adversarial attack on the original form (left), and PseudoSaccade form (right). Image in the centre column is a visual representation of the adversarial attack. Observe that classification on the PseudoSaccades forms gives the correct prediction of the true label with low confidence scores.*

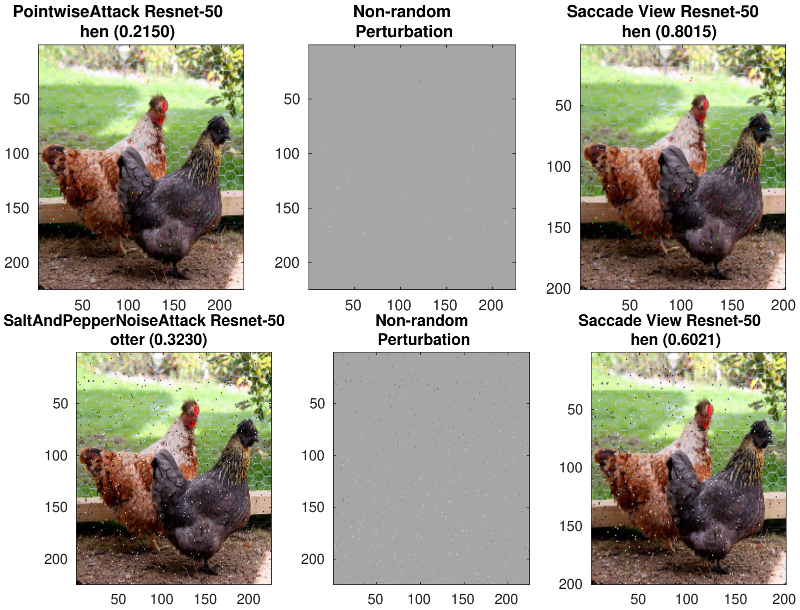

**Figure 7.8:** *ResNet-50 classification of "pixel-based" adversarial attacks on the original form (left), and PseudoSaccade form (right). Image in the centre column is a visual representation of the adversarial attack. Observe that classification is robust on the PseudoSaccades forms, returning prediction on the true label with moderate confidence scores.*

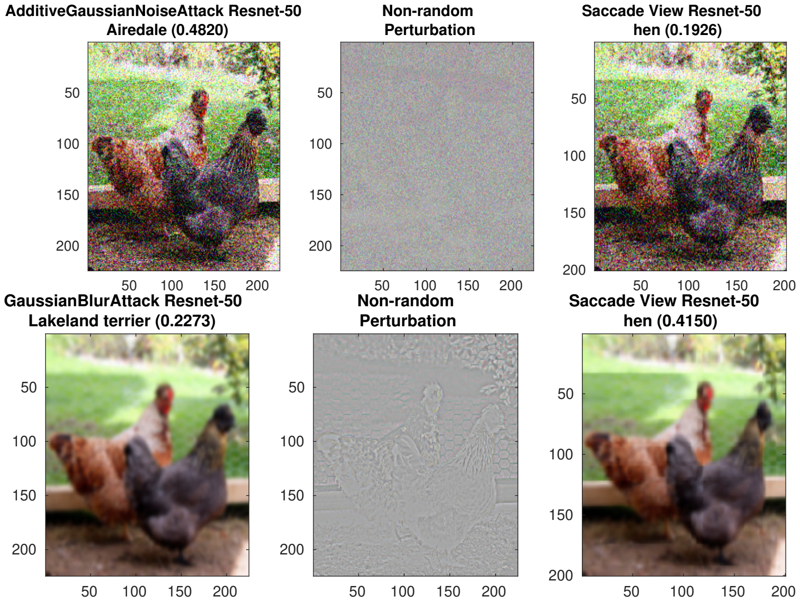

**Figure 7.9:** *ResNet-50 classification of "contrast-based" adversarial attacks on the original form (left), and PseudoSaccade form (right). Image in the centre column is a visual representation of the adversarial attack. Observe that classification is not robust on the PseudoSaccades forms, returning a low confidence prediction on the true label.*

We also demonstrated how PseudoSaccades could be used to improve the robustness of the DNN against adversarial examples It would be interesting to investigate the robustness of PseudoSaccades to adversarial examples that are transferable to human beings (Elsayed et al., 2018). We conjecture that further research into PseudoSaccades might reveal deeper insight into the mechanism of human vision and suggest methods how future research into image recognition can leverage on the advantages of these mechanisms while mitigating the disadvantages.

An open problem is whether a (simple or low overhead) non-uniform sampling scheme for constructing PseudoSaccades data exists that could improve performance further, possibly mediated by a scene-dependent prior. But it looks like a hard problem, in particular how to construct such a prior, although human visual processing suggests that such a scheme should be at least a possibility. We are examining non-uniform sampling schemes such as stratified sampling, and also techniques such as seam-carving, with a view to progress in this direction.

# 8

# Conclusion and Future Direction

We began this research by trying to gain insights into the open question "When will an ensemble of weak learners outperform a single carefully-tuned learner?" (Durrant, 2013; Brown et al., 2005). We focused on ensemble classifiers in the high-dimensional settings using random subspaces and made inroads into some of the questions we raised in Section 1.2.

In Chapter 4, we derived the data-dependent conditions for norm-preservation in random subspace projections. We defined a measure $c$ and $c'$ based on the $\ell_\infty$ and $\ell_4$ norms of the data, which describes the 'lucky structure' that helps with the norm-preservations guarantees on RS projections.

We empirically demonstrated that datasets with low $c$ and $c'$ measures gives better norm-preservation performance with RS than on datasets with larger measures. We also empirically corroborated our theories using real-world high-dimensional data from different settings namely natural images, natural audio and sparse binary vectors. We demonstrated the degradation in norm preservation when the regularity conditions are not met and how it affects the norm-preservation performance of RS projections. Guided from our theory, we also showed that random subspace with non-uniform sampling which reduces the within-sample variance (i.e. stratified sampling) could improve norm-preservation performance.

We discussed the implications of our theory for classification by adapting the proof technique of Arriaga and Vempala (1999), for classification with a margin. Finally, we also discussed the implications of our theory on compressive sensing, namely sparse signal reconstruction, and as an aside, demonstrated how RS can be used instead of a dense sensing matrix in reconstructing image data.

In Chapter 5, we showed that a random subspace projection is equivalent to using a sub-gaussian projection matrix and the corresponding sub-Gaussian norm and exploiting a lemma in Kabán and Durrant (2017), we derived the upper bound on the generalization error of the compressive ERM classification. We also derived data-dependent upper bounds on the flipping probability, and demonstrated how the flipping probability is related to the $\ell_4, \ell_6$ and $\ell_\infty$ norms and how these norms are an analogue to the regularity constant from chapter 4.

Guided by our theory, we proposed a computationally efficient method to reduce the flipping probability using a computationally efficient "densification" algorithm using Householder transformations. We empirically corroborated our results and discussed why our bound are still pessimistic. We also discussed the implications of this result on classification ensembles and discuss how our results show that we can recover the Bayes' classifier asymptotically using an RS ensemble of ERM classifiers.

In Chapter 6, we demonstrated how we could model a majority vote ensemble when the errors in the classifiers are not independent using a Polya-Eggenberger distribution. We showed how the diversity measure $\rho$ as described by Sneath and Sokal (1963) recovers the dispersion parameter of the distribution. We also empirically compared several diversity measures used in literature and discussed briefly how we may be able to estimate the diversity measures a priori. We also discussed the implications of our model and give a plausible explanation for the efficacy of ensemble methods like Random Forest based on the findings of our theory. We analysed the error based on our model and presented a 'good' and 'bad diversity' ambiguity decomposition similar to that of Brown and Kuncheva (2010) and discussed its limitations.

We empirically corroborated our findings on both synthetic and real-world data and discussed the limitations and the assumptions of our models. We demonstrated how our theory predicted the performance of RS ensembles and potential future research directions based on our findings.

We also discussed the sum-rule combination scheme and the error decomposition for the combination scheme. We reconciled the findings of Schapire (1990) and Blum (1997) to apparently contradictory finding of Kuncheva and Rodríguez (2014) and showed that a weighted voting scheme improves the accuracy of a classification ensemble more, in particular that a weighted scheme on a dataset with high feature noise could give improved ensemble performance over equally weighted members. We also showed that RS ensembles also work as a form of regularization for linear classifiers, similar to RP ensembles (Durrant and Kabán, 2014), although our approach differs considerably from theirs.

We demonstrated in Chapter 7 that using a very simple, and computationally cheap, 'PseudoSaccades' ensemble learning approach based on RS could improve the image classification performance of DNNs. This improvement is small but statistically significant at the 5% level and requires no retraining of the neural network. Following a careful analysis of the sources of error in our classification problem, we showed that these improvements also propagate to a weighted ensemble of PseudoSaccades versions of (off-the-shelf) DNNs.

We also demonstrated how PseudoSaccades could be used to improve the robustness of DNNs against adversarial examples. We speculate that further research into PseudoSaccades could provide new insights into the mechanisms of human visual processing and may suggest methods for future research into image recognition that can leverage on the advantages in the mechanism of human vision.

In this thesis, we focused primarily on random dimensionality reduction as a diversity generator and it could be informative to extend this research to use Bagging (Breiman, 1996) and Boosting (Schapire and Freund, 2012) in conjunction with the random subspace method. As noted in our discussion in section 6.3, our Polya-Eggenberger model suggests a plausible explanation for the effectiveness of Random Forests. Our theory also suggests that there may be data-dependent sampling scheme of the training features that would optimize the diversity-accuracy trade-offs in the learning the classifiers.

In our analysis, we have focused primarily on 2-class classifications. As the Polya-Eggenberger distribution also belongs to a Dirichlet-multinomial distribution, it could be possible to generalize our results to model the accuracies of some of the combination schemes in a $m$-class classification (e.g. Plurality vote or Borda Count).

In our analysis, we used Cantelli's inequality as an approximation for the cumulative distribution function (CDF) of the Polya-Eggenberger distribution. As noted in section 6.3.3, this bound is quite loose and sometimes has misleading implications. While we have yet to find a closed form for the Generalized Hypergeometric Function in CDF, replacing the error decomposition with the closed form of the CDF may reveal further insights on the accuracy of a majority vote ensemble classifier.

It is also known that that the weights used for weighted majority votes assumes independence in the classifiers (Section 4.3.3 of Zhou (2012)) and therefore modelling the dependence may lead to improved performance. Here, the diversity measure $\rho$ of the weighted ensemble could be described by an expression of the form

$$\frac{1}{\sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j} \sum_{i=1}^{N} \sum_{j\neq i}^{N} \frac{w_i w_j P_{ij} - \bar{p}^2}{\bar{p}(1-\bar{p})}$$

where $\bar{p} = \frac{\sum_{i=1}^{N} w_i p_i}{\sum_{i=1}^{N} w_i}$. It may be possible (using quadratic programming or some optimization algorithm) to find an optimal set of $w_i$ such that the loss of the ensemble given $\rho$ and $\overline{p}$ is minimized.

One shortcoming of using the Sneath and Sokal (1963) diversity measure is that we need the individual classifier performance in order to estimate the parameters of the Polya-Eggenberger model. Diversity estimates based on the outputs of the member classifiers (e.g. vote correlation), or the structure of the member classifiers (e.g. Jaccard-similarity index) do not appear to give consistent estimate of the parameter. It may also be possible to use the flipping probability discussed in chapter 5 to provide an estimate of the diversity measure.

Another open problem is whether a (simple or low overhead) non-uniform sampling scheme for constructing PseudoSaccades data exists that could improve performance further, possibly mediated by a scene-dependent prior. Human visual processing suggests that such a scheme should be at least a possibility. We are examining non-uniform sampling schemes such as stratified sampling, and also techniques such as seam-carving, with a view to progress in this direction.

It would also be interesting to investigate the robustness of PseudoSaccades to adversarial examples that are transferable to human beings (Elsayed et al., 2018). Intuitively we would expect that PseudoSaccades to also have problems with these transferable adversarial examples however understanding the underlying reason behind the difficulty may reveal insights into the similarities (and dissimilarities) between the mechanisms in machine vision and human vision.

We have stopped short of using RS data in training neural networks e.g. for data augmentation in training sets with small number of training examples or as hardening against adversarial examples (Madry et al., 2017), although we note an obvious parallel with drop-out regularization; The promising results from PseudoSaccades suggests that there may be some gains to be had from using PseudoSaccades as a form of data-augmentation.

# A

# Proof of Theorems

We will use the following two lemmas which are from Hoeffding (1963); Serfling (1974). Recall from Chapter 3

**Corollary A.1** (to Lemma 3.5, Hoeffding (1963) Section 6.)**.** *Let $C := c_1, c_2, \ldots, c_d$ be a finite population of $d$ values where $\forall j = 1, 2, \ldots, d$ we have $c_j \in [a_j, b_j]$ with probability 1. Let $X_i$ and $Y_i$, $i = 1, 2, \ldots, k$ be samples without and with replacement from $C$ respectively and define by $S_k(X)$ and $S_k(Y)$ the corresponding sample totals. Fix $t > 0$. Then it holds that:*

$$Pr\{|S_k(X) - E[S_k(X)]| \geq t\} \leq Pr\{|S_k(Y) - E[S_k(Y)]| \geq t\}$$

Note that $\mathrm{E}[S_k(X)] = \mathrm{E}[S_k(Y)]$, thus we may bound the probability of a large deviation in the sample total from its expectation in the case of a (non-independent) sample without replacement by the corresponding probability for an independent sample with replacement.

**Lemma A.2** (Serfling (1974) Corollary 1.1.)**.** *Let $C := c_1, c_2, \ldots, c_d$ be a finite population of $d$ values where $\forall j = 1, 2, \ldots, d$ we have $c_j \in [a_j, b_j]$ with probability 1. Let $X_i$, $i = 1, 2, \ldots, k$ be a simple random sample without replacement from $C$. Denote by $S_k := \sum_{i=1}^{k} X_i$ and define the sampling fraction $f_k := (k-1)/d$. Fix $t > 0$. Then:*

$$Pr\{|S_k - E[S_k]| \geq t\} \leq 2\exp\left(-\frac{2t^2}{(1-f_k)\sum_{i=1}^{k}(b_i - a_i)^2}\right)$$

*Comment:* Since $1 - f_k = (d - k + 1)/d < 1$ Lemma A.2 gives a strictly tighter bound than Lemma 3.5 for sampling without replacement, but brings in a dependence on $d$. We note that bounds for sampling without replacement which are somewhat tighter than those in Serfling (1974) when $k \simeq d$ were recently proved in Bardenet and Maillard (2015), in particular an empirical variant for when the population parameters are unknown. In our proof each population is a fixed vector of known length where the data dimension $d$ is the population size and the projection dimension $k$ is the sample size; thus in our setting we have access to both the full population and its parameters.

## A.1 Proof of Basic Bound

We prove the basic bound using Lemma 3.5 and Corollary A.1 and our without replacement bound then follows directly. The basic idea is to treat each vector as a finite population of size $d$ and RS as a simple random sample of size $k$ without replacement from it in the above lemmas, and then follow the line of argument in the usual proof of the JLL.

Let $X \in \mathbb{R}^d$ be an arbitrary, but fixed, real-valued vector and without loss of generality let $\|X\|_2^2 = 1$ (since otherwise we can take $X = Z/\|Z\|_2$). Denote by $X^2 := (X_1^2, X_2^2, \ldots, X_d^2)^T$ the vector containing the squared components of $X$. Assume without loss of generality that $\|X^2\|_\infty \leq \frac{c}{d}\|X\|_2^2$.

Now let $P \in \mathcal{M}_{d\times d}$ be a projection onto $k$ standard coordinate vectors, where the projection basis is chosen by sampling uniformly at random from all $\binom{d}{k}$ possible such bases. As noted already in Subsection 2.1.9 this is mathematically equivalent to an RS projection. Then in every random $P$ it holds that $k$ of the $P_{ii} = 1$ and every other entry of $P$ is zero so $\mathrm{Tr}(P) = k$ for any $P$, and therefore $\mathrm{Tr}(\mathrm{E}[P]) = \mathrm{E}[\mathrm{Tr}(P)] = k$. Furthermore, since $\Pr\{P_{ii} = p\} = \Pr\{P_{jj} = p\}$ for all $i, j \in \{1, 2, \ldots, d\}$ and $p \in \{0, 1\}$, it follows that $\mathrm{E}[P_{ii}] = \mathrm{E}[P_{jj}] = k/d, \forall i, j$ by symmetry. Thus $\mathrm{E}[P] = \frac{k}{d}I$ and $\mathrm{E}[\|PX\|_2^2] = \frac{k}{d}\|X\|_2^2$, where both expectations are taken with respect to the random draws of $P$ and we used the fact that $P^TP = PP = P, \forall P$.

We want to upper bound the following probability:

$$\Pr\left\{\left|\frac{d}{k}\|PX\|_2^2 - \|X\|_2^2\right| \geq \epsilon\right\} = \Pr\left\{\left|\frac{d}{k}\|PX\|_2^2 - \frac{d}{k}\mathrm{E}\left[\|PX\|_2^2\right]\right| \geq \epsilon\right\}$$

We give details for one side of the inequality using the basic Hoeffding bound, the other cases proceed along the same lines. Now, for any fixed instance of $P$ denote by

$I$ the index set such that $i \in I \iff P_{ii} = 1$. Then:

$$\Pr\left\{\|PX\|_2^2 \geq \frac{k}{d}\epsilon + \mathrm{E}\left[\|PX\|_2^2\right]\right\} = \Pr\left\{\sum_{i\in I} X_i^2 \geq \frac{k}{d}\left(\epsilon + \sum_{i=1}^{d} X_i^2\right)\right\}$$

where the sample total $\sum_{i\in I} X_i^2$ is estimated from a sample of size $k$ without replacement. Applying Lemma 3.5 and Corollary A.1 we then have:

$$\begin{aligned}&\Pr\left\{\textstyle\sum_{i\in I} X_i^2 \geq \frac{k}{d}\left(\epsilon + \sum_{i=1}^{d} X_i^2\right)\right\}\\ &= \Pr\left\{\tfrac{d}{k}\|PX\|_2^2 - \|X\|_2^2 \geq \epsilon\right\} \leq \exp\left(-\frac{2k\left(\frac{\epsilon}{d}\right)^2}{\|X^2\|_\infty^2}\right)\end{aligned}$$

The lower bound proceeds similarly and yields the same probability guarantee for a single fixed vector:

$$\Pr\left\{\|X\|_2^2 - \frac{d}{k}\|PX\|_2^2 \geq \epsilon\right\} \leq \exp\left(-\frac{2k\left(\frac{\epsilon}{d}\right)^2}{\|X^2\|_\infty^2}\right)$$

Thus by union bound, and using the condition on the theorem $\|X^2\|_\infty \leq \frac{c}{d}\|X\|_2^2$ to kill the unwanted dependence on $d$, we obtain the following guarantee for an arbitrary unit-norm vector $X$:

$$\Pr\left\{\left|\|X\|_2^2 - \frac{d}{k}\|PX\|_2^2\right| \geq \epsilon\right\} \leq 2\exp\left(-\frac{2k\epsilon^2}{c^2\|X\|_2^4}\right) \tag{A.1}$$

To complete the proof we consider a set, $\mathcal{T}_N$, of $N$ vectors in $\mathbb{R}^d$ and let $X_i$ and $X_j$ be any two vectors in this set. Instantiating $X$ in A.1 as $(X_i - X_j)/\|X_i - X_j\|_2$ and then applying union bound again over all $\binom{N}{2} < N^2/2$ inter-point distances in $\mathcal{T}_N$ we obtain, for all pairs $X_i, X_j \in \mathcal{T}_N$ simultaneously, it holds that:

$$\Pr\left\{\left|\|X_i - X_j\|_2^2 - \frac{d}{k}\|PX_i - PX_j\|_2^2\right| \geq \epsilon\right\} \leq N^2\exp\left(-\frac{2k\epsilon^2}{c^2}\right)$$

Where we substituted $\|X\|_2^4 = 1$ in RHS. Finally, setting the probability upper bound on the RHS to $\delta$ and solving for $k$ gives the theorem.

For the without replacement bound, one simply follows the same steps as above, but using the Serfling bound (Lemma A.2) in place of the Hoeffding bound (Lemma 3.5), finally setting the RHS to $\delta$ and solving for $k/1 - f_k$ to complete the proof.

## A.2 Proof of Bernstein-Bennett Bound

As in the proof of the basic bound, let $X \in \mathbb{R}^d$ be an arbitrary, but fixed, real-valued vector and let $P$ be a projection onto $k$ canonical basis vectors chosen randomly without replacement. Also let $Q$ be a diagonal random matrix with its non-zero entries chosen i.i.d as follows:

$$Q_{ii} := \begin{cases} 1 - \frac{k}{d} & \text{w.p. } \frac{k}{d} \\ -\frac{k}{d} & \text{otherwise.} \end{cases}$$

In what follows $Q$ will act as a 'proxy' for $P$: In particular we will show that $\|QX\|_2^2$ and $\|PX\|_2^2 - \frac{k}{d}\|X\|_2^2$ are related and have the same expectation, but $\text{Var}\|QX\|_2^2 \geq \text{Var}\|PX\|_2^2$, and therefore we can use $Q$ to obtain a bound on a quantity involving $P$. First, we note that:

$$\mathrm{E}\left[\sum_{i=1}^{d} Q_{ii}x_i^2\right] = \sum_{i=1}^{d}\left(1-\frac{k}{d}\right)x_i^2\left(\frac{k}{d}\right) + \left(-\frac{k}{d}\right)x_i^2\left(1-\frac{k}{d}\right) = 0 \qquad (2)$$

and also:

$$\mathrm{E}\left[\|PX\|_2^2 - \frac{k}{d}\|X\|_2^2\right] = \sum_{i=1}^{d} x_i^2\frac{k}{d} - \frac{k}{d}x_i^2 = \mathrm{E}\left[\|QX\|_2^2\right]$$

Furthermore:

$$\begin{aligned}\text{Var}\left[\sum_{i=1}^{d} Q_{ii}x_i^2\right] &= \mathrm{E}\left[\left(\sum_{i=1}^{d} Q_{ii}x_i^2\right)^2\right] - \mathrm{E}\left[\sum_{i=1}^{d} Q_{ii}x_i^2\right]^2 \\ &= \sum_{i=1}^{d}\mathrm{E}[Q_{ii}^2]x_i^4 + \sum_{i=1}^{d}\sum_{j\neq i}^{d}\mathrm{E}[Q_{ii}x_i^2]\mathrm{E}[Q_{jj}x_j^2] = \sum_{i=1}^{d}\mathrm{E}[Q_{ii}^2]x_i^4 \\ &= \sum_{i=1}^{d}\left(1-\frac{k}{d}\right)\frac{k}{d}x_i^4 = \frac{dk-k^2}{d^2}\|X\|_4^4\end{aligned}$$

$$\begin{aligned}\text{Var}\left[\|PX\|_2^2 - \frac{k}{d}\|X\|_2^2\right] =&\mathrm{E}\left[\left(\|PX\|_2^2 - \frac{k}{d}\|X\|_2^2\right)^2\right] - \mathrm{E}\left[\|PX\|_2^2 - \frac{k}{d}\|X\|_2^2\right]^2 \\ =&\mathrm{E}\left[\left(\sum_{i=1}^{d}\left(P_{ii}-\frac{k}{d}\right)x_i^2\right)^2\right] \\ =&\mathrm{E}\left[\sum_{i=1}^{d}\left(P_{ii}^2 - \frac{2k}{d}P_{ii} + \frac{k^2}{d^2}\right)x_i^4\right. \\ &\left.+\sum_{i=1}^{d}\sum_{j\neq i}^{d}\left(P_{ii}P_{jj} - \frac{k}{d}P_{ii} - \frac{k}{d}P_{jj} + \frac{k^2}{d^2}\right)x_i^2x_j^2\right] \\ =&\sum_{i=1}^{d}\left(\frac{dk-k^2}{d^2}\right)x_i^4 + \sum_{i=1}^{d}\sum_{j\neq i}^{d}\left(\frac{k(k-1)}{d(d-1)} - \frac{k^2}{d^2}\right)x_i^2x_j^2 \\ =&\frac{dk-k^2}{d^2}\|X\|_4^4 - \left(\frac{dk-k^2}{d^2(d-1)}\right)\sum_{i=1}^{d}x_i^2\left(\|X\|_2^2 - x_i^2\right) \\ \leq&\frac{dk-k^2}{d^2}\|X\|_4^4\end{aligned}$$

Since by $\|X\|_2^2 > x_i^2$, Var $\left[\sum_{i=1}^d Q_{ii}x_i^2\right] \geq$ Var $\left[\|PX\|_2^2 - \frac{k}{d}\|X\|_2^2\right]$

$$\begin{aligned}\Pr\left\{\left|\frac{d}{k}\|PX\|_2^2 - \|X\|_2^2\right| > \epsilon\|X\|_2^2\right\} =&\Pr\left\{\left|\|PX\|_2^2 - \frac{k}{d}\|X\|_2^2\right| > \frac{k}{d}\epsilon\|X\|_2^2\right\} \\ \leq&\Pr\left\{\left|\sum_{i=1}^d Q_{ii}x_i^2\right| > \frac{k}{d}\epsilon\|X\|_2^2\right\}\end{aligned}$$

Now let $t > 0$, we have:

$$\begin{aligned}\mathrm{E}\left[\exp\left[t\sum_{i=1}^d Q_{ii}x_i^2\right]\right] = \mathrm{E}&\left[\sum_{n=0}^{\infty}\frac{t^n}{n!}\left(\sum_{i=1}^d Q_{ii}x_i^2\right)^n\right] \\ =&1 + \sum_{n=2}^{\infty}\frac{t^n}{n!}\mathrm{E}\left[\left(\sum_{i=1}^d Q_{ii}x_i^2\right)^2\left(\sum_{i=1}^d Q_{ii}x_i^2\right)^{n-2}\right] \quad \text{using } \textit{equation } 2 \\ \leq&1 + \sum_{n=2}^{\infty}\frac{t^n}{n!}\mathrm{E}\left[\left(\sum_{i=1}^d Q_{ii}x_i^2\right)^2\left(\frac{d-k}{d}\|X\|_2^2\right)^{n-2}\right] \\ =&1 + \frac{k\|X\|_4^4}{(d-k)\|X\|_2^4}\sum_{n=2}^{\infty}\frac{(t(d-k))^n}{d^n n!}\|X\|_2^{2n} \\ =&1 + \frac{k\|X\|_4^4}{(d-k)\|X\|_2^4}\left(\exp\left[\frac{t(d-k)}{d}\|X\|_2^2\right] - 1 - \frac{t(d-k)}{d}\|X\|_2^2\right) \\ \leq&\exp\left[\frac{k\|X\|_4^4}{(d-k)\|X\|_2^4}\left(\exp\left[\frac{t(d-k)}{d}\|X\|_2^2\right] - 1 - \frac{t(d-k)}{d}\|X\|_2^2\right)\right]\end{aligned}$$

Where the first inequality comes from observing that $(\sum_{i=1}^d Q_{ii}x_i^2) \leq \frac{d-k}{d}\|X\|_2^2$ and the second uses the inequality $1 + x \leq e^x, \forall x \geq 0$ . We therefore have:

$$\begin{aligned}&\Pr\left\{\sum_{i=1}^d Q_{ii}x_i^2 > \frac{k}{d}\epsilon\|X\|_2^2\right\} \leq \min_{t>0}\frac{\mathrm{E}\left[\exp\left[t\sum_{i=1}^d Q_{ii}x_i^2\right]\right]}{\exp\left[t\frac{k}{d}\epsilon\|X\|_2^2\right]} \\ &\leq \min_{t>0}\exp\left[-\frac{tk\epsilon\|X\|_2^2}{d} + \frac{k\|X\|_4^4}{(d-k)\|X\|_2^4}\left(\exp\left[\frac{t(d-k)\|X\|_2^2}{d}\right] - 1 - \frac{t(d-k)\|X\|_2^2}{d}\right)\right] \quad (3)\end{aligned}$$

Choosing $t = \frac{d}{(d-k)\|X\|_2^2}\log\left(1 + \frac{\epsilon\|X\|_2^4}{\|X\|_4^4}\right)$ and substituting in equation 3, we have:

$$\begin{aligned}\Pr\left\{\sum_{i=1}^d Q_{ii}x_i^2 > \frac{k}{d}\epsilon\|X\|_2^2\right\} &\leq \exp\left[-\frac{k\|X\|_4^4}{(d-k)\|X\|_2^4}\left[\left(1 + \frac{\epsilon\|X\|_2^4}{\|X\|_4^4}\right)\log\left(1 + \frac{\epsilon\|X\|_2^4}{\|X\|_4^4}\right) - \frac{\epsilon\|X\|_2^4}{\|X\|_4^4}\right]\right] \\ &= \exp\left[-\frac{k\|X\|_4^4}{(d-k)\|X\|_2^4}\phi\left(\frac{\epsilon\|X\|_2^4}{\|X\|_4^4}\right)\right] \quad (4)\end{aligned}$$

Where $\phi(x) = (1+x)log(1+x) - x$

Substituting using $\phi(x) = (1+x)log(1+x) - x > \frac{x^2}{2+\frac{2}{3}x}$ in equation 4, we have:

$$\begin{aligned}\Pr\left\{\sum_{i=1}^d Q_{ii}x_i^2 > \frac{k}{d}\epsilon\|X\|_2^2\right\} \leq& \exp\left[-\frac{k\|X\|_4^4}{(d-k)\|X\|_2^4}\frac{\epsilon^2\|X\|_2^8}{\|X\|_4^8}\frac{1}{2 + \frac{2\epsilon\|X\|_2^4}{3\|X\|_4^4}}\right] \\ =& \exp\left[-\frac{k\epsilon^2\|X\|_2^4}{2(d-k)\left(\|X\|_4^4 + \frac{1}{3}\epsilon\|X\|_2^4\right)}\right] \\ \leq& \exp\left[-\frac{k\epsilon^2\|X\|_2^4}{4d\max(\|X\|_4^4, \frac{\epsilon}{3}\|X\|_2^4)}\right] =: \delta\end{aligned}$$

Substituting $Q_{ii}$ with $-Q_{ii}$ gives the lower bound in a similar fashion, with the same failure probability $\delta$. Choosing $c'^2 > \frac{8d\|X\|_4^4}{\|X\|_2^4}$. The proof is complete using the condition on the theorem and union bounding to give:

$$\Pr\left\{\left|\|X\|_2^2 - \frac{d}{k}\|PX\|_2^2\right| \geq \epsilon\|X\|_2^2\right\} \leq 2\exp\left(-\frac{2k\epsilon^2}{c'^2}\right) \quad \Box$$

# B

# Regularity Constants for Some Distributions

We first note that $\|X\|_4^4 = \sum_{i=1}^d x_i^4$ and $\|X\|_2^2 = \sum_{i=1}^d x_i^2$. With this, we can estimate the "contributions" to the $\ell_4$ and $\ell_2$ norms of each of the $x_i$ by calculating the expected values of $\mathrm{E}[x_i^4]$ and $E[x_i^2]$ respectively. Note that $E[x_i^4]$ is the non-central fourth norm (kurtosis) and $\mathrm{E}[x_i^2]$ is the non-central second norm (variance + mean$^2$), and these values are known for many distributions. Some distributions are unbounded, giving an unbounded $\|X\|_\infty^2$. In these cases, we set $\|x_i\|_\infty$ to the mean plus 4 times the standard deviation $\bar{x}_i + 4\sigma(x_i)$. For ease of calculations, we normalized $x_i$ to have $\mathrm{E}[x_i^2] = 1$.

| Distribution | $\mathrm{E}[\|x_i^2\|_\infty]$ | $\mathrm{E}[x_i^4]$ |
|---|---|---|
| Normal with mean 0, variance 1 | 16 | 3 |
| Scaled Bernoulli | $\frac{1}{p}$ | $\frac{1}{p}$ |
| Scaled Rademacher | 1 | 1 |
| Scaled Chi Squared k=1 | $\frac{49}{3} + \frac{8}{\sqrt{3}}$ | $\frac{105}{9}$ |
| Continuous uniform distribution $(0, \sqrt{3})$ | 3 | 9/5 |
| Continuous uniform distribution $(-\sqrt{3}, \sqrt{3})$ | 3 | 9/5 |
| Poisson | $\sim 3.77$ | $\frac{3}{2}(1+\sqrt{5})$ |
| Triangular Distribution $(\sqrt{6}, 0, \sqrt{6})$ | 6 | $\frac{12}{5}$ |

**Table B.1:** *Estimated $E[\|x_i^2\|_\infty]$ & $E[x_i^4]$ for commonly used distributions*

# C

# Householder Transforms

## C.1 Densification Algorithm

Let $\boldsymbol{v}$ be a 'dense' vector (i.e. $\boldsymbol{v} = \frac{1}{\sqrt{d}}(\pm 1, \dots, \pm 1)$). We will give an algorithm to find $\boldsymbol{H} = \boldsymbol{I}_d - 2\boldsymbol{n}\boldsymbol{n}^T$ such that $\boldsymbol{H}\boldsymbol{x} = \boldsymbol{v}$.

Let $\boldsymbol{v} = \boldsymbol{H}\boldsymbol{x}$. Observe that $\boldsymbol{v} = \boldsymbol{x} - 2\boldsymbol{n}\boldsymbol{n}^T\boldsymbol{x} = \boldsymbol{x} - 2\boldsymbol{n}\cos\beta$. Observe also that $\boldsymbol{n}^T\boldsymbol{x} = \cos\beta = \sin(\alpha/2)$ as shown in figure C.1. Using the half angle formula for sin, this implies that $\boldsymbol{n}^T\boldsymbol{x} = \sqrt{\frac{1-\cos\alpha}{2}} = \sqrt{\frac{1-\boldsymbol{x}^T\boldsymbol{v}}{2}}$. Therefore,

$$\begin{aligned} \boldsymbol{v} =& \boldsymbol{x} - 2\boldsymbol{n}\sqrt{\frac{1-\boldsymbol{x}^T\boldsymbol{v}}{2}} \\ \implies \boldsymbol{n} =& \frac{\boldsymbol{x}-\boldsymbol{v}}{\sqrt{2(1-\boldsymbol{x}^T\boldsymbol{v})}} \end{aligned}$$

Note that the algorithm can also be used to reflect $\boldsymbol{x}$ to any arbitrary vector $\boldsymbol{v}$.

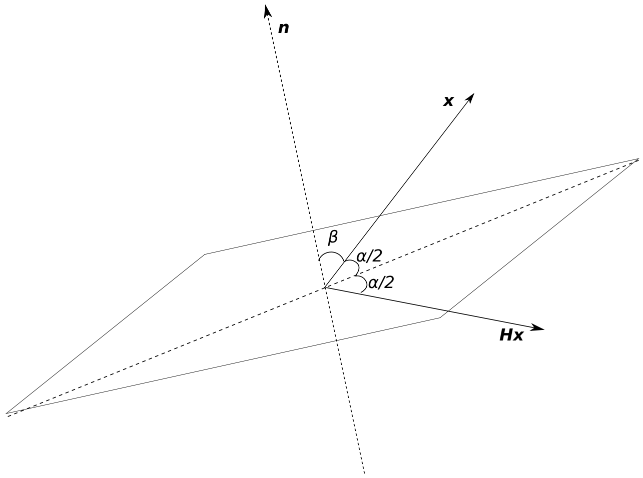

**Figure C.1:** *The hyperplane of reflection and the relationship between the angles between the normal vector to the hyperplane, the 'reflected' vector, the 'original' vector and the hyperplane*

```
$d \leftarrow$ dimensionality of the vector
Normalize $\boldsymbol{x} \leftarrow \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}$
Randomly generate $\boldsymbol{v}$
    $\boldsymbol{v}_i := \begin{cases} \frac{1}{\sqrt{d}} & \text{w.p } \frac{1}{2} \\ -\frac{1}{\sqrt{d}} & \text{w.p } \frac{1}{2} \end{cases}$
Let $c \leftarrow \sqrt{2(1 - \boldsymbol{x}'^T \boldsymbol{v})}$, if $c = 0$, regenerate $\boldsymbol{v}$
for $i \leftarrow 1$ to $d$ do
    - Let $\boldsymbol{n}_i \leftarrow \frac{\boldsymbol{x}'_i - \boldsymbol{v}_i}{c}$
end for
Normalize $\boldsymbol{n} \leftarrow \frac{\boldsymbol{n}}{\|\boldsymbol{n}\|}$
```

**Algorithm C.1:** *Densification Algorithm using Householder Transform.*

## C.2 Orthogonal Vector Generation

In the following, we want to generate $d-1$ mutually orthogonal vectors to vector $\boldsymbol{x}$. This algorithm can be used to generate an orthogonal basis vectors such that $\boldsymbol{x}$ is one of the orthogonal basis vectors.

Observe that reflection preserve the angular separation in vectors, as shown in lemma 5.8. Therefore, a Householder's transform of an orthogonal basis (i.e. the usual basis) results in another orthogonal basis. Choose $\boldsymbol{v} = \boldsymbol{e}^{(1)}$. We then find the Householder transform, that reflects $\boldsymbol{Hx} = \boldsymbol{v}$ using algorithm C.1. Observe that the columns of $(\boldsymbol{HI})^T = \boldsymbol{H}^T$ is an orthogonal basis, with $\boldsymbol{x}$ the first column.

## C.3 Orthogonalization

Householder transforms can also be used for triangularization and orthogonalization of matrices. The advantage of using Householder Transforms for orthogonalization over traditional methods of orthogonalization such as Gram-Schmidt is that using Householder transforms for orthogonalization gives better orthogonality (i.e. $\boldsymbol{Q}^T\boldsymbol{Q} \approxeq \boldsymbol{I}$ ).

The following algorithm is taken from Stewart (1998), and the Matlab implementation of this algorithm is provided by Moler (2016).

```
$\boldsymbol{v} \leftarrow \boldsymbol{e}^{(1)}$
for $k \leftarrow 1$ to $\min(p, q)$ do
        - $\boldsymbol{x} \leftarrow \boldsymbol{X}_{k:n,k}$
        - $mR_{k,k} \leftarrow \|\boldsymbol{x}\|$,
        - Normalize $\boldsymbol{x} \leftarrow \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}$.
        - Find $\boldsymbol{n}$ using algorithm C.1 such that $\boldsymbol{H}\boldsymbol{x} = \boldsymbol{v}$
        - $\boldsymbol{U}_{k:p,k} \leftarrow \boldsymbol{n}_{k:p}$
        - $\boldsymbol{w} \leftarrow \boldsymbol{n}_{k,p}^T \boldsymbol{X}_{k:p,k+1:q}$
        - Update $\boldsymbol{X}_{k:p,k+1:q} \leftarrow \boldsymbol{X}_{k:p,k+q:q} - \boldsymbol{n}_{k:p}\boldsymbol{w}$
        - Update $\boldsymbol{R}_{k,k+1:q} \leftarrow \boldsymbol{X}_{k,k+1:q}$
end for
```

**Algorithm C.2:** *Orthogonalization Algorithm using Householder Transform taken from Stewart (1998)*

# D

# Additional Figures and Tables

## D.1 Appendix to Chapter 5

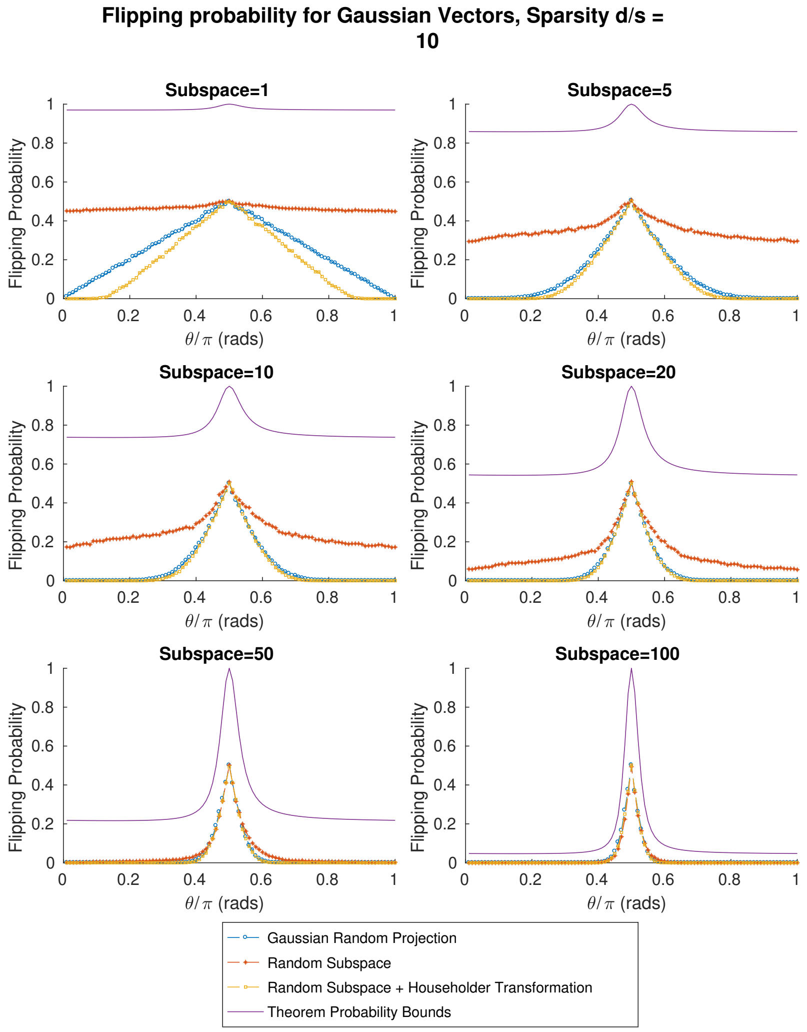

**Figure D.1:** *Flipping probability vs angular separation for two Gaussian vectors, with sparsity $s = 10$ for projection dimension $k \in \{1, 5, 10, 20, 50, 100\}$ and dimensionality $d = 1000$.*

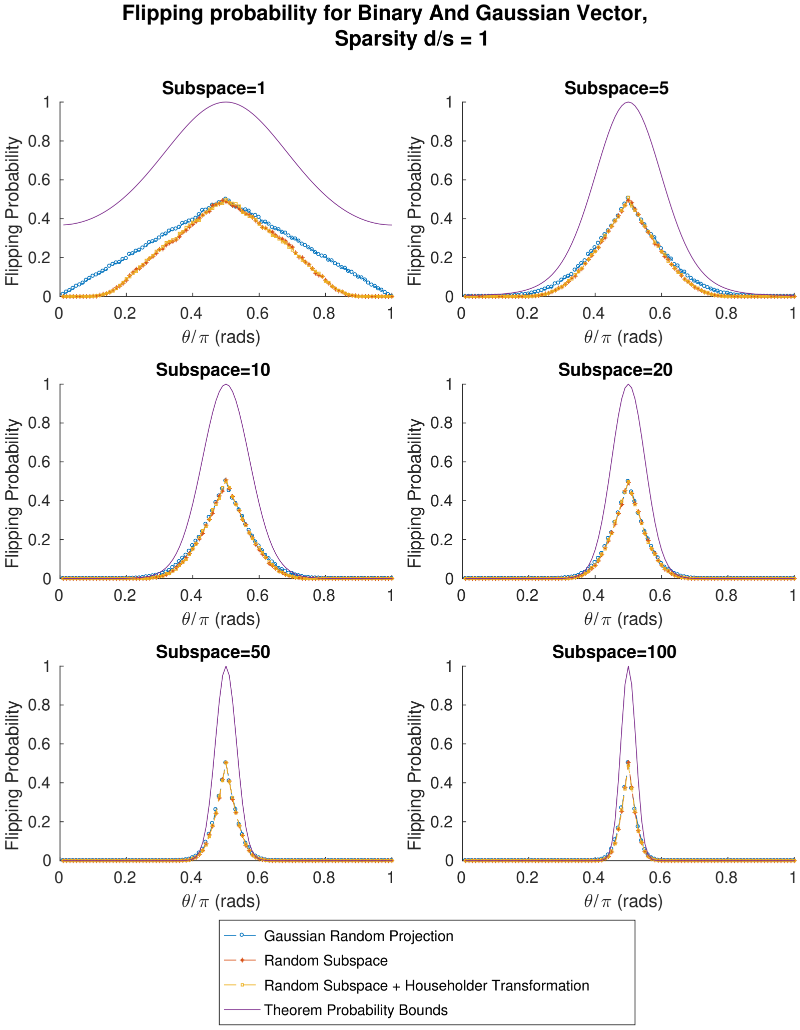

**Figure D.2:** *Flipping probability vs angular separation for a Binary vector and a Gaussian vector, with sparsity $s = 1$ for projection dimension $k \in \{1, 5, 10, 20, 50, 100\}$ and dimensionality $d = 1000$.*

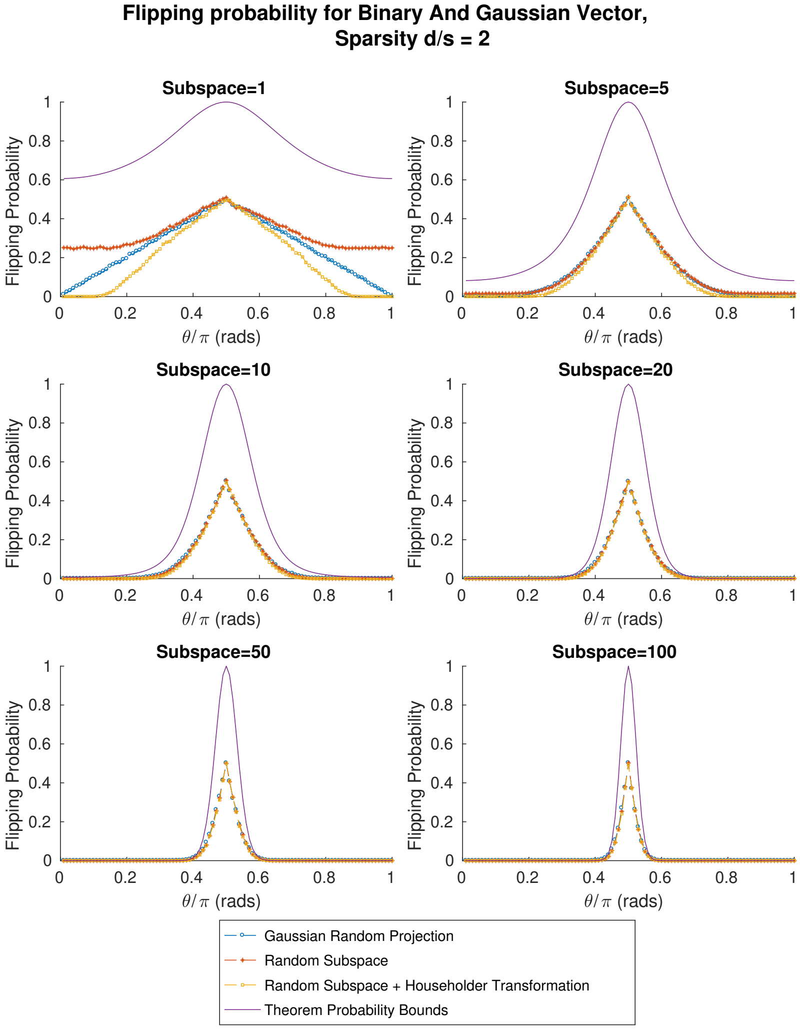

**Figure D.3:** *Flipping probability vs angular separation for a Binary vector and a Gaussian vector, with sparsity $s = 2$ for projection dimension $k \in \{1, 5, 10, 20, 50, 100\}$ and dimensionality $d = 1000$.*

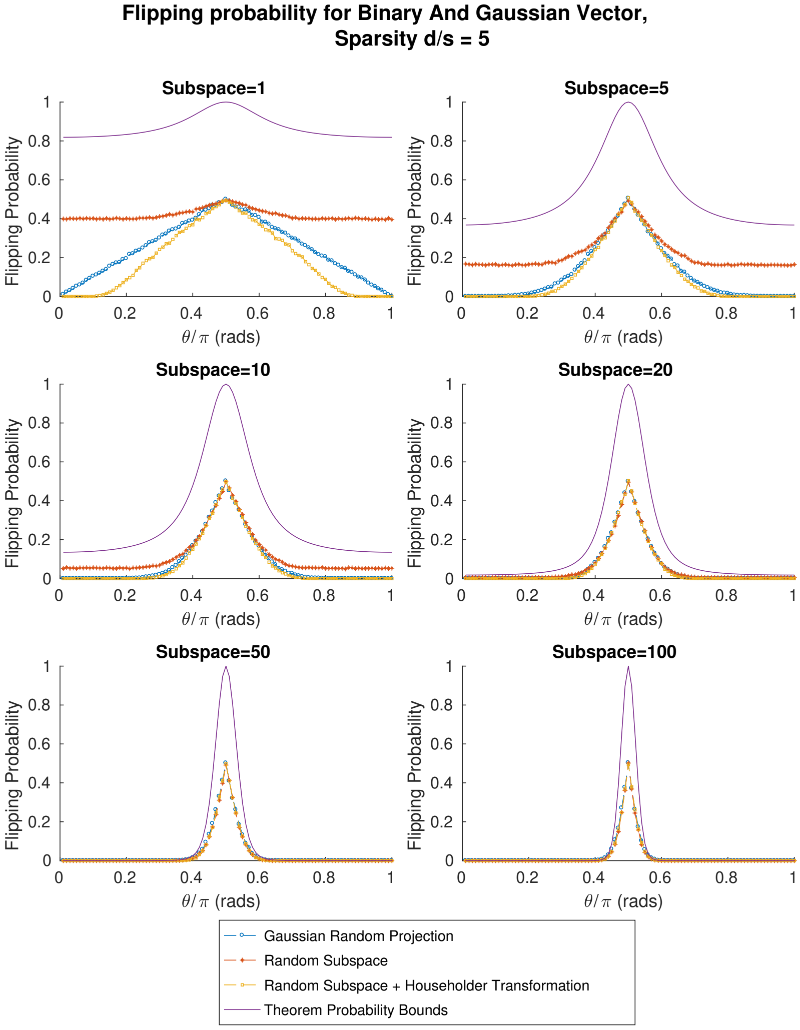

**Figure D.4:** *Flipping probability vs angular separation for a Binary vector and a Gaussian vector, with sparsity $s = 5$ for projection dimension $k \in \{1, 5, 10, 20, 50, 100\}$ and dimensionality $d = 1000$.*

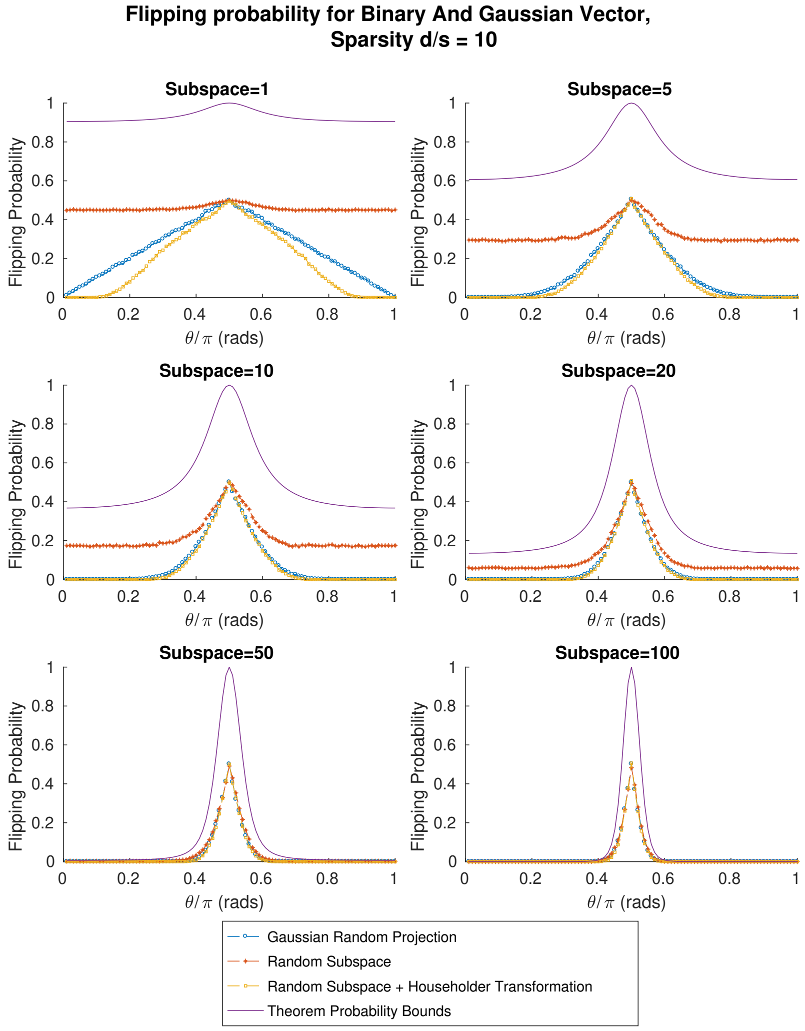

**Figure D.5:** *Flipping probability vs angular separation for a Binary vector and a Gaussian vector, with sparsity* $s = 10$ *for projection dimension* $k \in \{1, 5, 10, 20, 50, 100\}$ *and dimensionality* $d = 1000$.

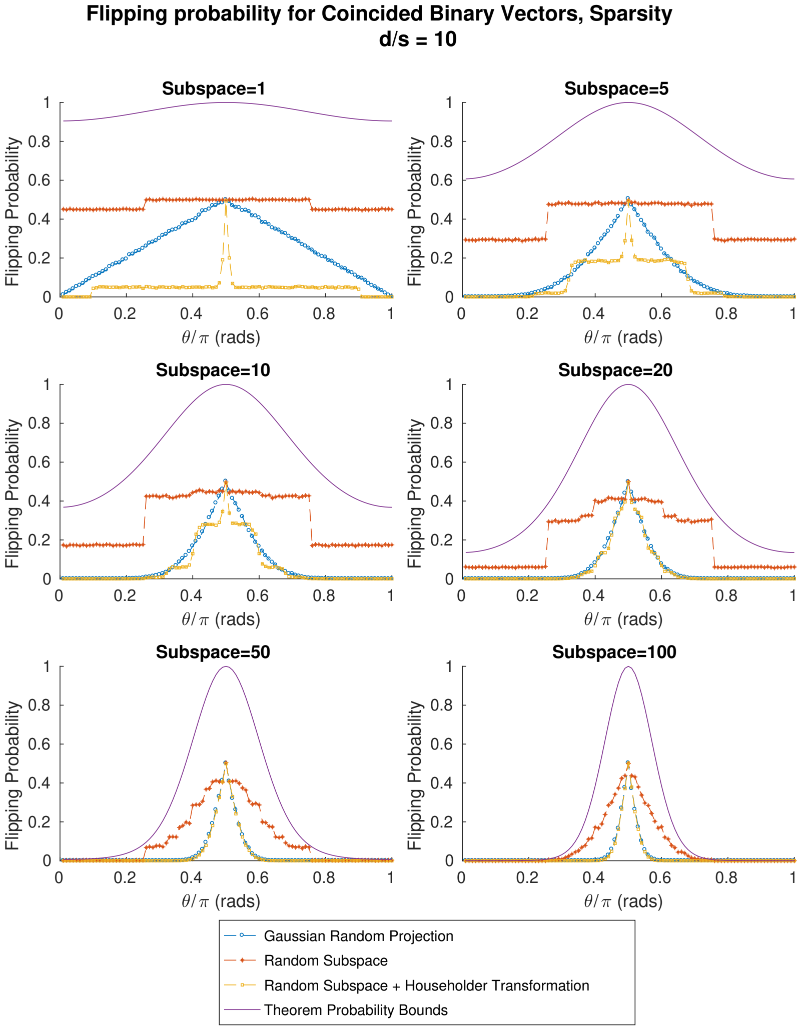

**Figure D.6:** *Flipping probability vs angular separation for two Binary Vector such that the two vectors coincide in every coordinate with sparsity $s = 10$ for projection dimension $k \in \{1, 5, 10, 20, 50, 100\}$ and dimensionality $d = 1000$.*

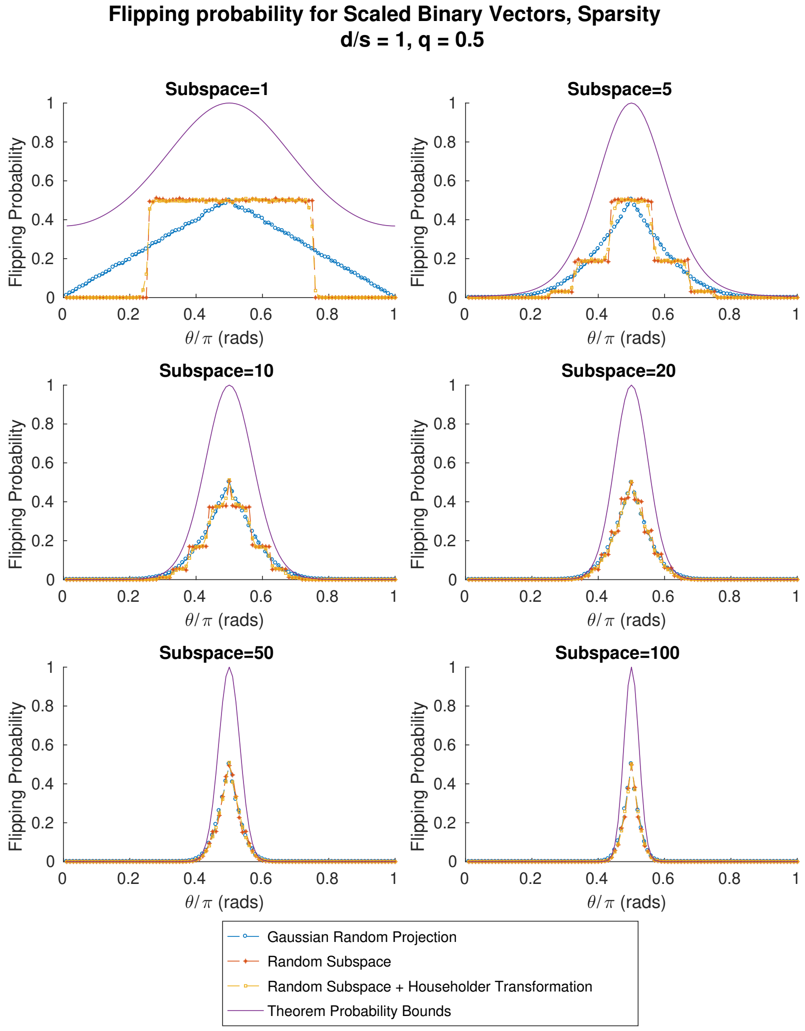

**Figure D.7:** *Flipping probability vs angular separation for two Binary Vector such that the coincidence in every coordinate is 1/2 with sparsity $s = 1$ for projection dimension $k \in \{1, 5, 10, 20, 50, 100\}$ and dimensionality $d = 1000$.*

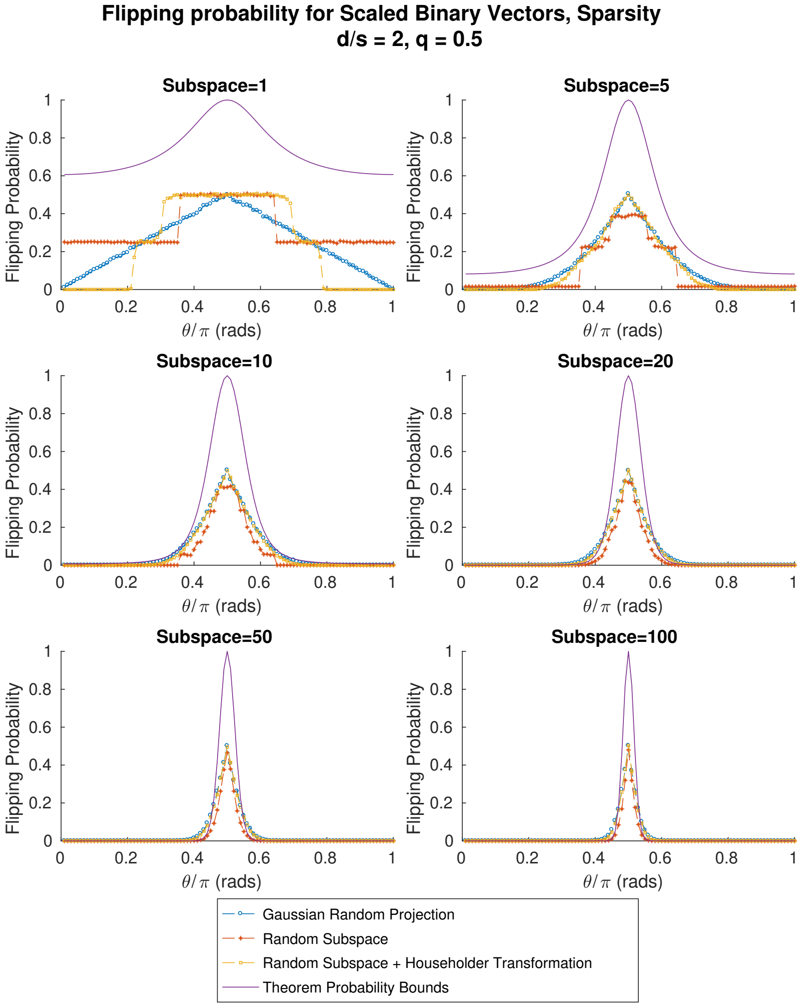

**Figure D.8:** *Flipping probability vs angular separation for two Binary Vector such that the coincidence in every coordinate is 1/2 with sparsity $s = 2$ for projection dimension $k \in \{1, 5, 10, 20, 50, 100\}$ and dimensionality $d = 1000$.*

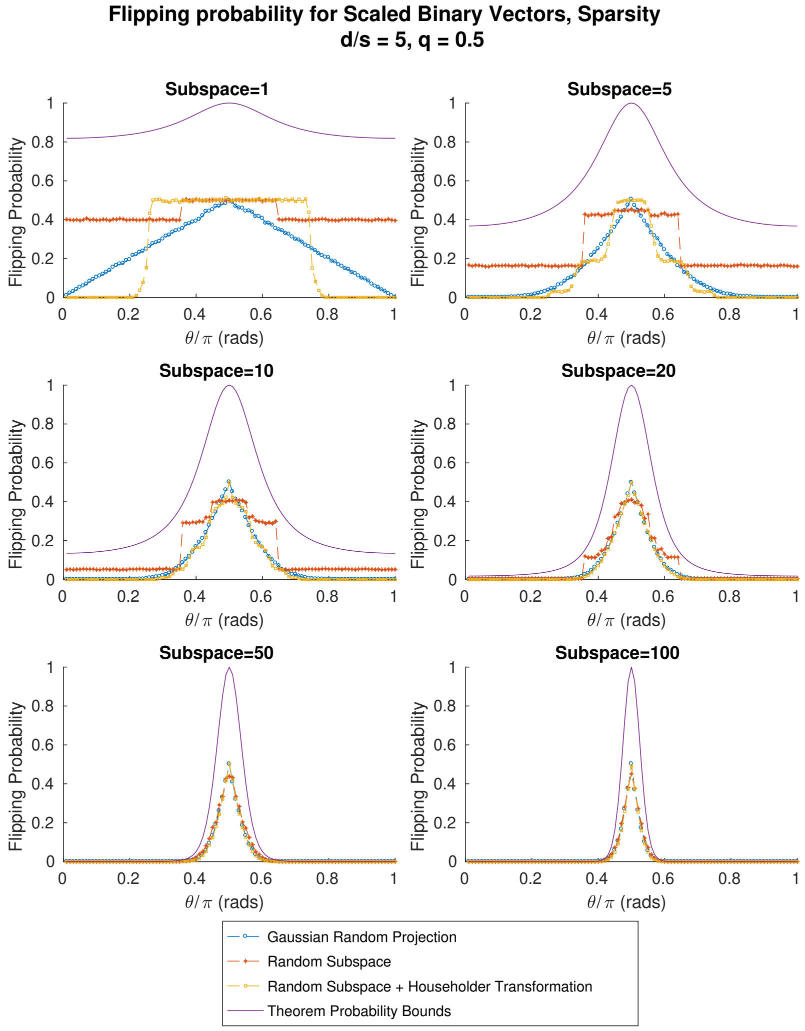

**Figure D.9:** *Flipping probability vs angular separation for two Binary Vector such that the coincidence in every coordinate is 1/2 with sparsity $s = 5$ for projection dimension $k \in \{1, 5, 10, 20, 50, 100\}$ and dimensionality $d = 1000$.*

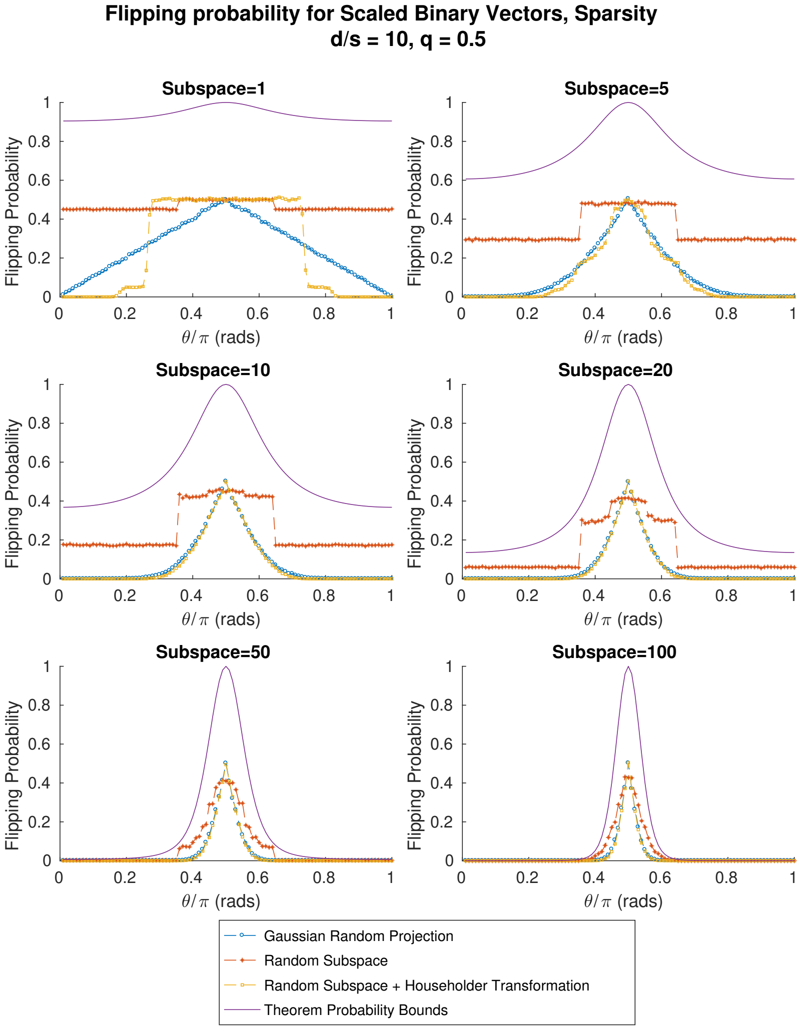

**Figure D.10:** *Flipping probability vs angular separation for two Binary Vector such that the coincidence in every coordinate is 1/2 with sparsity $s = 10$ for projection dimension $k \in \{1, 5, 10, 20, 50, 100\}$ and dimensionality $d = 1000$.*

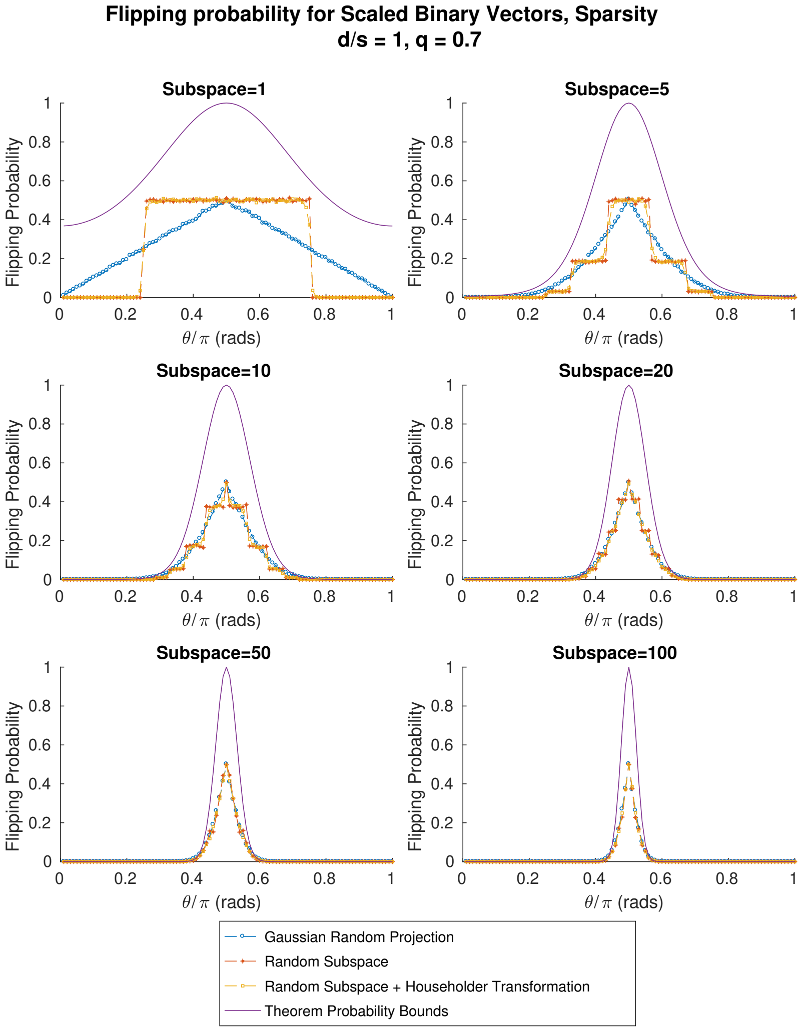

**Figure D.11:** *Flipping probability vs angular separation for two Binary Vector such that the coincidence in every coordinate is 2/3 with sparsity $s = 1$ for projection dimension $k \in \{1, 5, 10, 20, 50, 100\}$ and dimensionality $d = 1000$.*

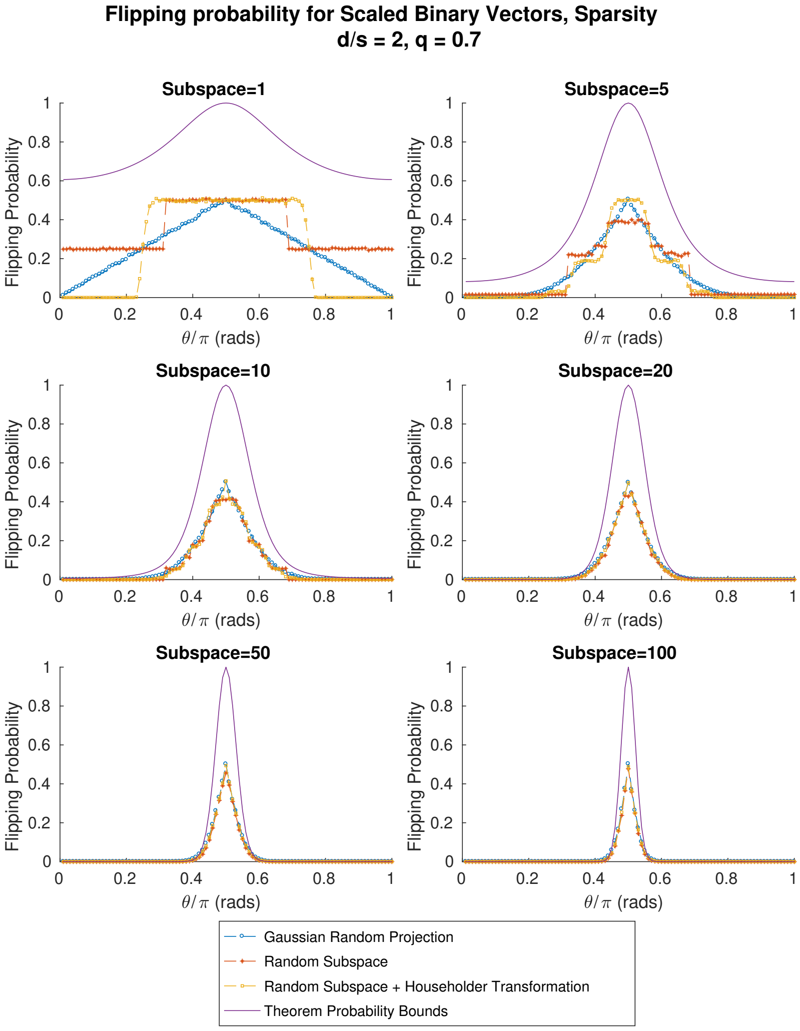

**Figure D.12:** *Flipping probability vs angular separation for two Binary Vector such that the coincidence in every coordinate is 2/3 with sparsity* $s = 2$ *for projection dimension* $k \in \{1, 5, 10, 20, 50, 100\}$ *and dimensionality* $d = 1000$*.*

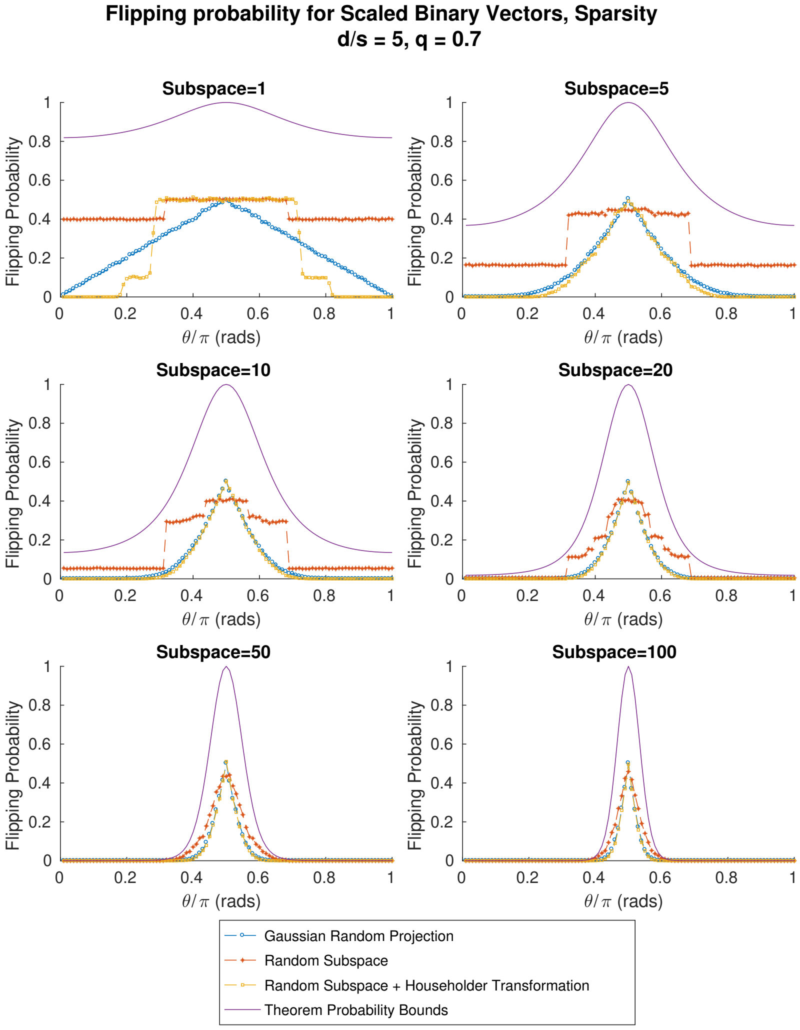

**Figure D.13:** *Flipping probability vs angular separation for two Binary Vector such that the coincidence in every coordinate is 2/3 with sparsity* $s = 5$ *for projection dimension* $k \in \{1, 5, 10, 20, 50, 100\}$ *and dimensionality* $d = 1000$.

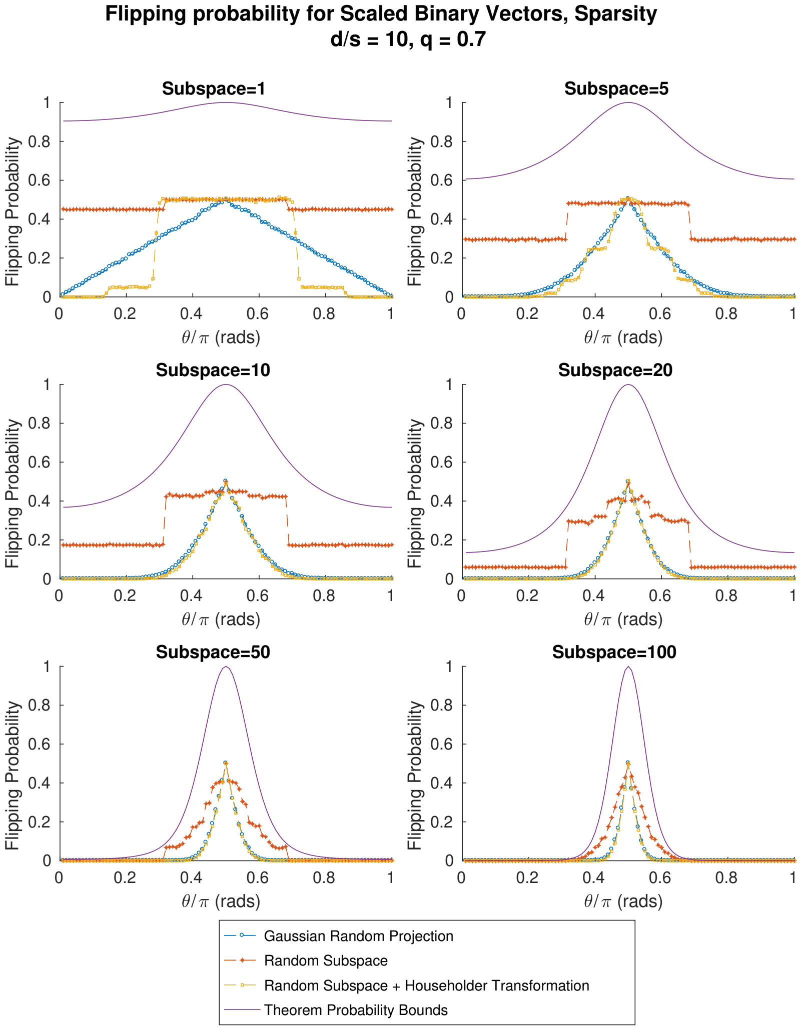

**Figure D.14:** *Flipping probability vs angular separation for two Binary Vector such that the coincidence in every coordinate is 2/3 with sparsity* $s = 10$ *for projection dimension* $k \in \{1, 5, 10, 20, 50, 100\}$ *and dimensionality* $d = 1000$.

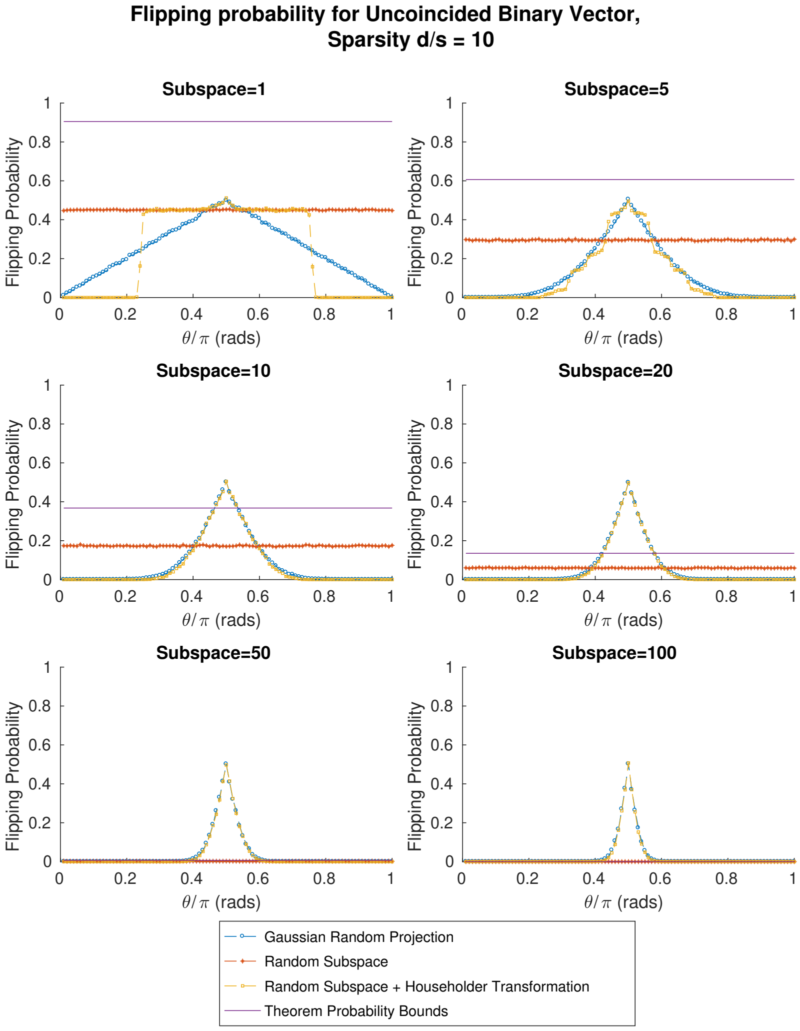

**Figure D.15:** *Flipping probability vs angular separation for two Binary Vector such that the two vectors do not coincidence in every coordinate with sparsity $s = 10$ for projection dimension $k \in \{1, 5, 10, 20, 50, 100\}$ and dimensionality $d = 1000$.*

## D.2 Appendix to Chapter 6

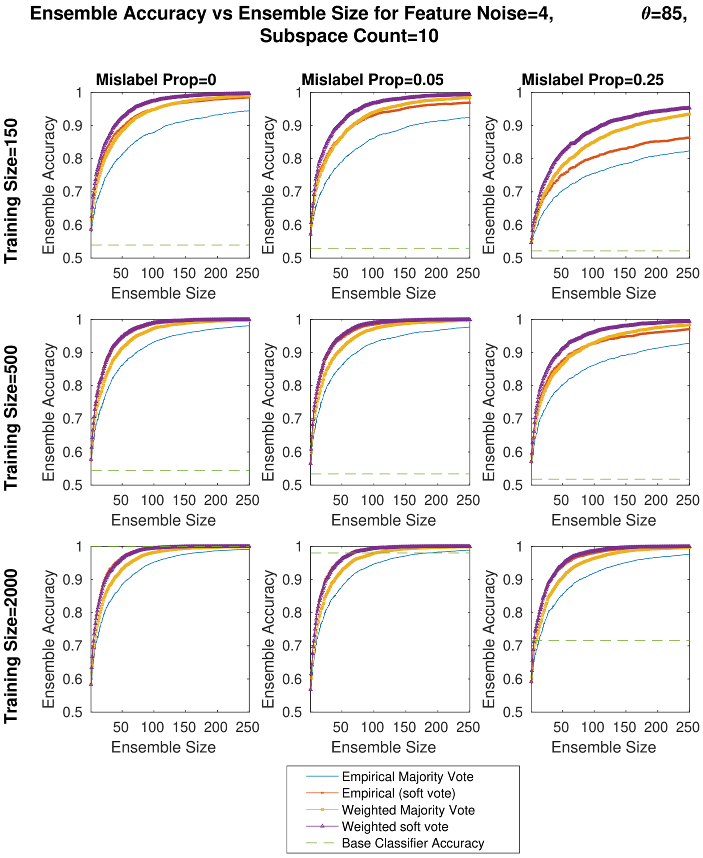

**Figure D.16:** *Ensemble classification accuracy vs ensemble member size for varying training size and mislabelling proportion, with feature noise $s = 4$, difficulty $\theta = 85$ and projection dimensions $k = 10$ and dimensionality $d = 1000$*

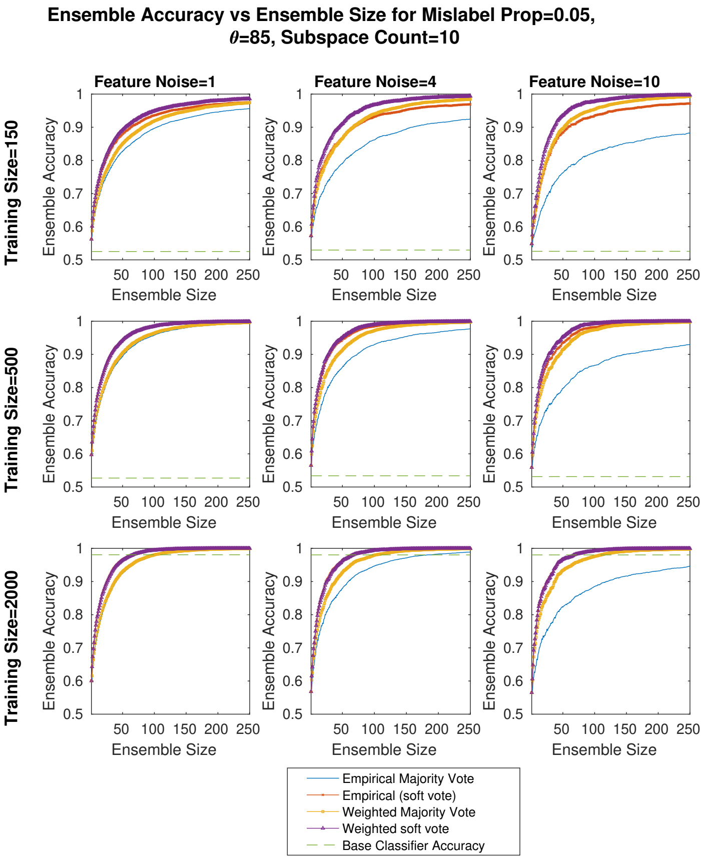

**Figure D.17:** *Ensemble classification accuracy vs ensemble member size for varying training size and feature noise, with mislabelling proportion $q = 0.05$, difficulty $\theta = 85$ and projection dimensions $k = 10$ and dimensionality $d = 1000$*

| | | k=10 | | | |
|---|---|---|---|---|---|
| theta | n | N=50 | N=100 | N=150 | N=250 |
| 80 | 150 | 99.4/99.5 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 |
| | 500 | 99.8/99.8 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 |
| | 2000 | 99.8/99.9 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 |
| 85 | 150 | 84.7/84.6 | 91.3/91.4 | 94.4/94.3 | 96.9/96.9 |
| | 500 | 90.6/90.3 | 96.2/96.2 | 98.3/98.3 | 99.5/99.5 |
| | 2000 | 92.7/92.7 | 97.8/97.9 | 99.3/99.3 | 99.9/99.9 |
| 87.5 | 150 | 63.0/62.5 | 66.4/66.1 | 68.1/68.1 | 70.3/70.5 |
| | 500 | 68.7/68.8 | 74.1/74.2 | 77.3/77.4 | 81.0/81.0 |
| | 2000 | 74.6/74.5 | 81.8/81.7 | 85.9/85.9 | 90.6/90.7 |

**Table D.1:** *Comparison between the empirical majority vote ensemble accuracy (left) and our model (right) for data with no irrelevant features $s = 1$,and $k = 10$. Observe that the values are within 1% of the model.*

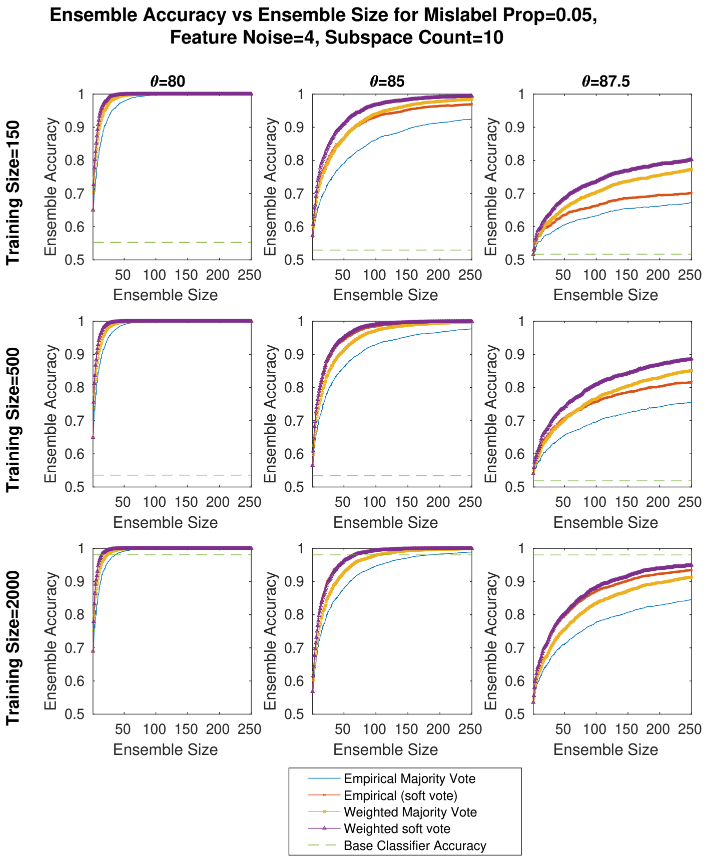

**Figure D.18:** *Ensemble classification accuracy vs ensemble member size for varying training size and difficulty, with mislabelling proportion $q = 0.05$, feature noise $s = 4$ and projection dimensions $k = 10$ and dimensionality $d = 1000$*

| | | k=50 | | | |
|---|---|---|---|---|---|
| theta | n | N=50 | N=100 | N=150 | N=250 |
| 80 | 150 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 |
| | 500 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 |
| | 2000 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 |
| 85 | 150 | 96.0/96.1 | 98.5/98.5 | 99.1/99.1 | 99.5/99.5 |
| | 500 | 99.6/99.6 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 |
| | 2000 | 99.9/99.9 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 |
| 87.5 | 150 | 70.0/70.0 | 72.9/73.1 | 74.6/74.4 | 75.5/75.7 |
| | 500 | 82.4/82.0 | 86.1/86.1 | 87.6/87.9 | 89.4/89.4 |
| | 2000 | 91.1/91.3 | 95.8/95.8 | 97.4/97.4 | 98.5/98.5 |

**Table D.2:** *Comparison between the empirical majority vote ensemble accuracy (left) and our model (right) for data with no irrelevant features $s = 1$, and $k = 50$. Observe that the values are within 1% of the model.*

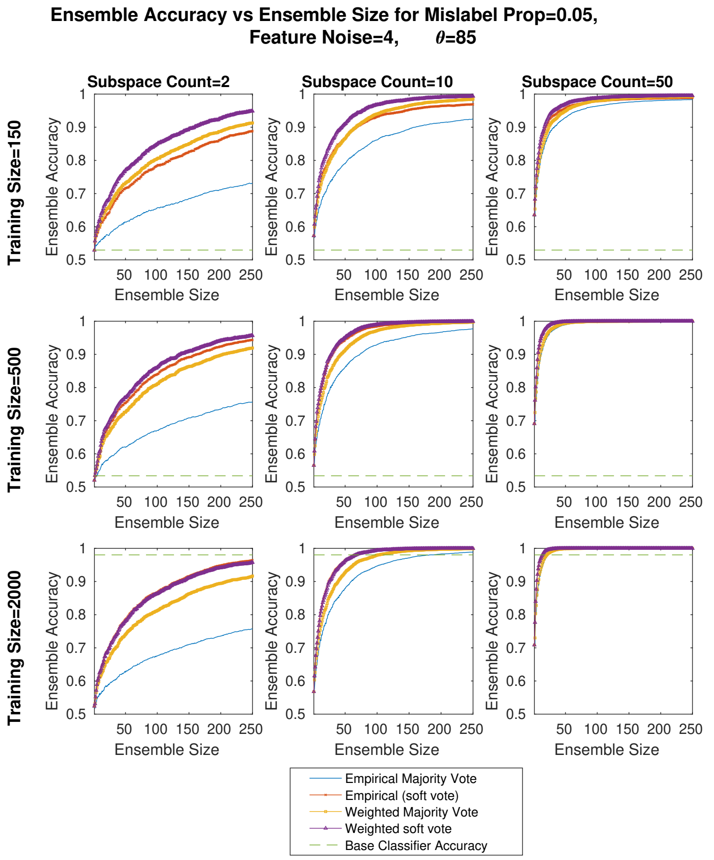

**Figure D.19:** *Ensemble classification accuracy vs ensemble member size for varying training size and projection dimensions, with mislabelling proportion $q = 0.05$, feature noise $s = 4$ and difficulty $\theta = 85$ and dimensionality $d = 1000$*

| | | k=2 | | | |
|---|---|---|---|---|---|
| theta | n | N=50 | N=100 | N=150 | N=250 |
| 80 | 150 | 74.7/75.0 | 82.7/82.5 | 87.0/86.9 | 91.9/91.7 |
| | 500 | 75.1/75.8 | 83.1/83.4 | 87.7/87.8 | 92.5/92.6 |
| | 2000 | 75.8/75.5 | 82.8/83.0 | 87.5/87.3 | 92.1/92.1 |
| 85 | 150 | 61.6/61.9 | 66.5/66.3 | 69.4/69.2 | 73.1/73.2 |
| | 500 | 62.6/63.3 | 67.6/68.2 | 71.2/71.4 | 75.8/75.8 |
| | 2000 | 63.4/63.5 | 68.3/68.3 | 71.4/71.6 | 76.0/76.0 |
| 87.5 | 150 | 54.4/54.5 | 56.3/56.2 | 57.3/57.4 | 59.1/59.1 |
| | 500 | 55.0/55.5 | 57.3/57.6 | 59.0/59.0 | 60.9/61.1 |
| | 2000 | 57.7/57.2 | 59.9/59.9 | 61.8/61.8 | 64.4/64.5 |

**Table D.3:** *Comparison between the empirical majority vote ensemble accuracy (left) and our model (right) for data with moderate $s = 4$ and $k = 2$. Observe that the values are within 1% of the model.*

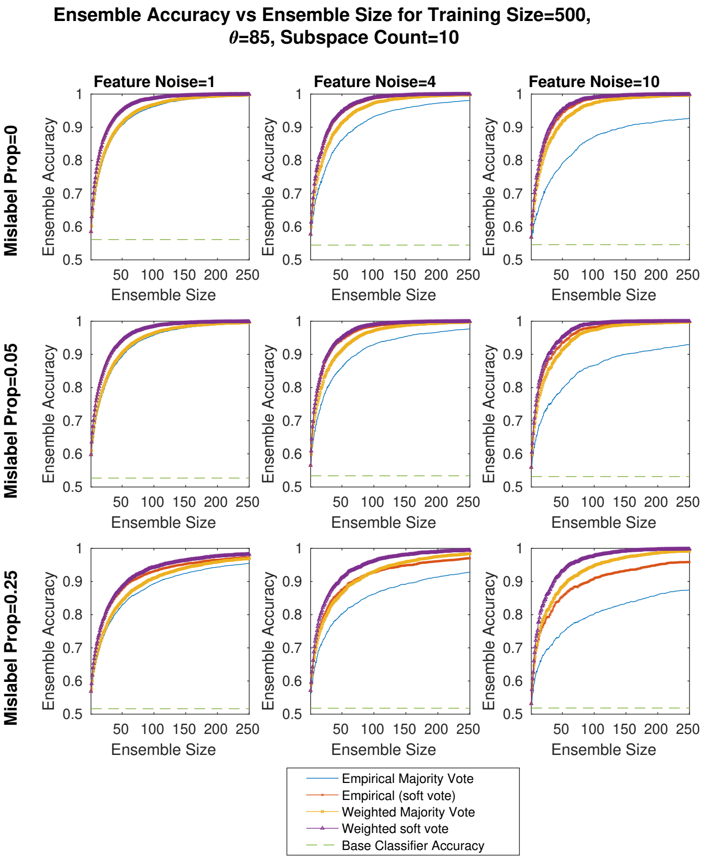

**Figure D.20:** *Ensemble classification accuracy vs ensemble member size for varying mislabelling proportion and feature noise, with training size $n = 500$, difficulty $\theta = 85$ and projection dimensions $k = 10$ and dimensionality $d = 1000$*

| | | k=10 | | | |
|---|---|---|---|---|---|
| theta | n | N=50 | N=100 | N=150 | N=250 |
| 80 | 150 | 98.2/98.4 | 99.8/99.8 | 100.0/100.0 | 100.0/100.0 |
| | 500 | 98.8/99.0 | 99.9/99.9 | 100.0/100.0 | 100.0/100.0 |
| | 2000 | 99.4/99.3 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 |
| 85 | 150 | 81.1/81.3 | 87.8/88.0 | 91.4/91.3 | 94.4/94.4 |
| | 500 | 86.0/86.6 | 93.1/93.2 | 95.8/96.0 | 98.1/98.1 |
| | 2000 | 88.4/88.8 | 95.2/95.2 | 97.6/97.5 | 99.1/99.1 |
| 87.5 | 150 | 61.4/61.5 | 64.8/64.8 | 66.8/66.7 | 68.9/68.9 |
| | 500 | 66.7/66.4 | 71.1/71.1 | 73.9/73.9 | 77.0/77.1 |
| | 2000 | 71.5/71.2 | 77.7/77.4 | 80.9/81.0 | 85.2/85.2 |

**Table D.4:** *Comparison between the empirical majority vote ensemble accuracy (left) and our model (right) for data with moderate $s = 4$ and $k = 10$. Observe that the values are within 1% of the model.*

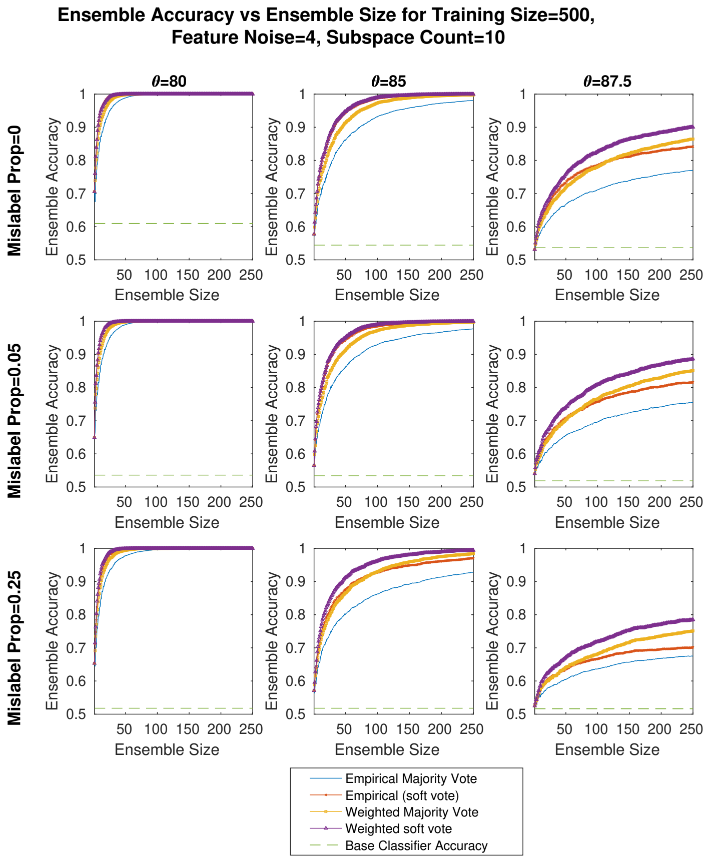

**Figure D.21:** *Ensemble classification accuracy vs ensemble member size for varying mislabelling proportion and difficulty, with training size $n = 500$, feature noise $s = 4$ and projection dimensions $k = 10$ and dimensionality $d = 1000$*

| | | k=50 | | | |
|---|---|---|---|---|---|
| theta | n | N=50 | N=100 | N=150 | N=250 |
| 80 | 150 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 |
| | 500 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 |
| | 2000 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 |
| 85 | 150 | 95.7/95.5 | 98.2/98.1 | 98.9/98.8 | 99.4/99.3 |
| | 500 | 99.3/99.4 | 99.9/99.9 | 100.0/100.0 | 100.0/100.0 |
| | 2000 | 99.9/99.9 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 |
| 87.5 | 150 | 69.3/69.5 | 72.2/72.5 | 73.9/73.8 | 75.1/75.1 |
| | 500 | 80.7/80.9 | 85.0/85.0 | 86.8/86.7 | 88.3/88.2 |
| | 2000 | 90.4/90.5 | 95.1/95.0 | 96.7/96.7 | 98.0/98.0 |

**Table D.5:** *Comparison between the empirical majority vote ensemble accuracy (left) and our model (right) for data with moderate $s = 4$ and $k = 50$. Observe that the values are within $1\%$ of the model.*

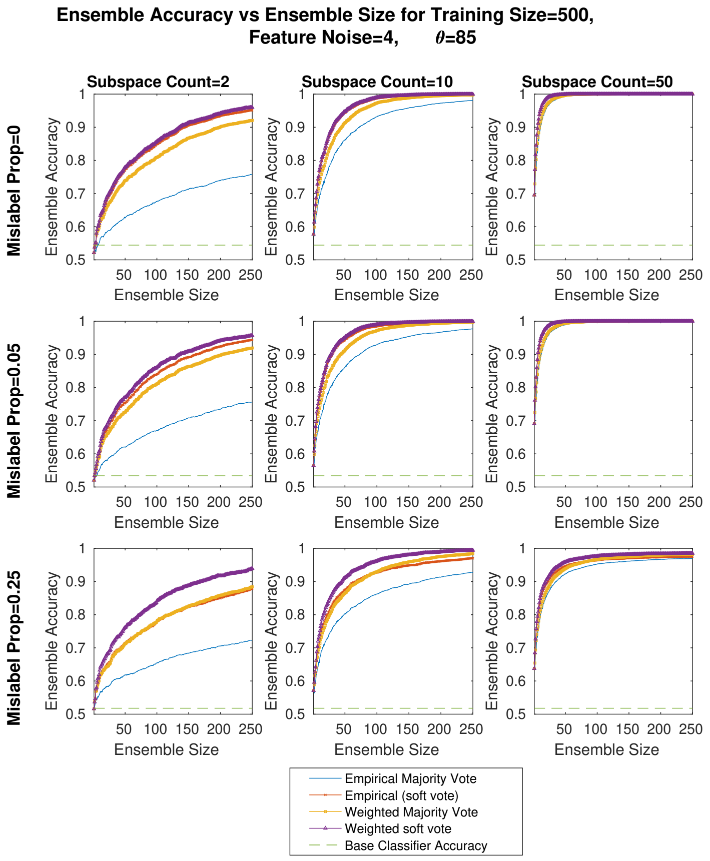

**Figure D.22:** *Ensemble classification accuracy vs ensemble member size for varying mislabelling proportion and projection dimensions, with training size $n = 500$, feature noise $s = 4$ and difficulty $\theta = 85$ and dimensionality $d = 1000$*

| | | k=10 | | | |
|---|---|---|---|---|---|
| theta | n | N=50 | N=100 | N=150 | N=250 |
| 80 | 150 | 94.5/94.2 | 98.0/98.1 | 99.1/99.1 | 99.7/99.7 |
| | 500 | 94.9/95.0 | 98.4/98.5 | 99.3/99.4 | 99.8/99.8 |
| | 2000 | 95.1/95.4 | 98.5/98.7 | 99.6/99.5 | 99.9/99.9 |
| 85 | 150 | 77.1/76.5 | 83.9/82.8 | 86.5/86.1 | 89.5/89.5 |
| | 500 | 79.0/79.6 | 87.1/86.2 | 90.2/89.5 | 92.7/92.7 |
| | 2000 | 81.7/81.5 | 88.4/88.3 | 91.6/91.5 | 94.7/94.6 |
| 87.5 | 150 | 60.5/60.5 | 63.5/63.4 | 65.0/65.1 | 67.1/67.1 |
| | 500 | 64.0/64.2 | 67.5/68.3 | 70.3/70.6 | 73.4/73.4 |
| | 2000 | 66.7/66.9 | 71.5/71.8 | 74.9/74.6 | 77.8/77.9 |

**Table D.6:** *Comparison between the empirical majority vote ensemble accuracy (left) and our model (right) for data with high $s = 10$ and $k = 10$. Observe that the values are within 1% of the model.*

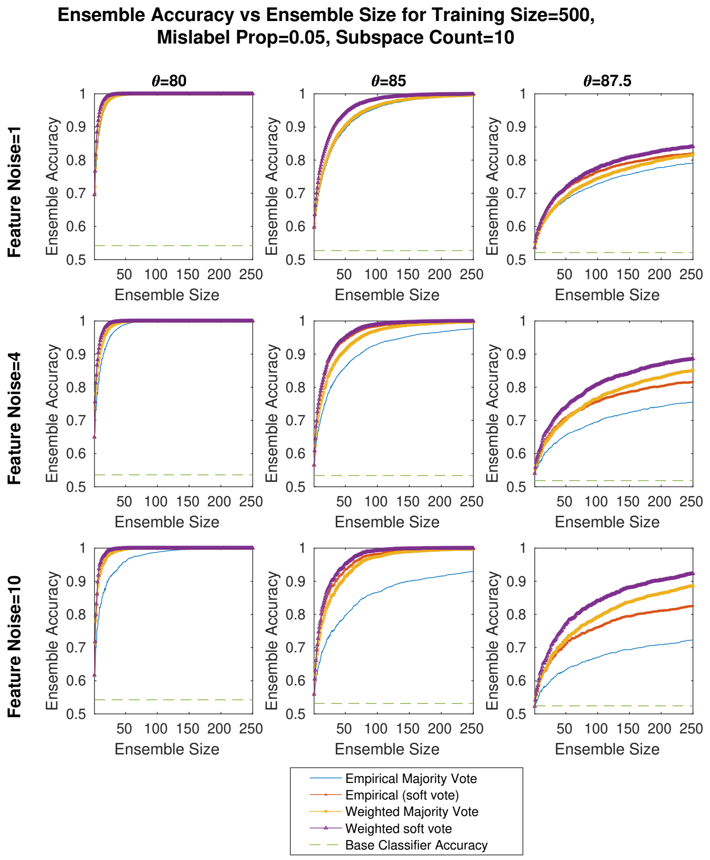

**Figure D.23:** *Ensemble classification accuracy vs ensemble member size for varying feature noise and difficulty, with training size $n = 500$, mislabelling proportion $q = 0.05$ and projection dimensions $k = 10$ and dimensionality $d = 1000$*

| | | k=50 | | | |
|---|---|---|---|---|---|
| theta | n | N=50 | N=100 | N=150 | N=250 |
| 80 | 150 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 |
| | 500 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 |
| | 2000 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 |
| 85 | 150 | 94.4/94.5 | 97.3/97.4 | 98.2/98.3 | 98.9/98.9 |
| | 500 | 98.8/98.7 | 99.8/99.7 | 99.9/99.9 | 100.0/100.0 |
| | 2000 | 99.5/99.6 | 100.0/100.0 | 100.0/100.0 | 100.0/100.0 |
| 87.5 | 150 | 69.2/69.1 | 72.1/72.0 | 73.2/73.3 | 74.6/74.5 |
| | 500 | 79.6/79.8 | 83.6/83.7 | 85.6/85.4 | 86.9/86.9 |
| | 2000 | 87.5/88.4 | 92.5/93.0 | 94.6/94.8 | 96.3/96.3 |

**Table D.7:** *Comparison between the empirical majority vote ensemble accuracy (left) and our model (right) for data with high $s = 10$ and $k = 50$. Observe that the values are within 1% of the model.*

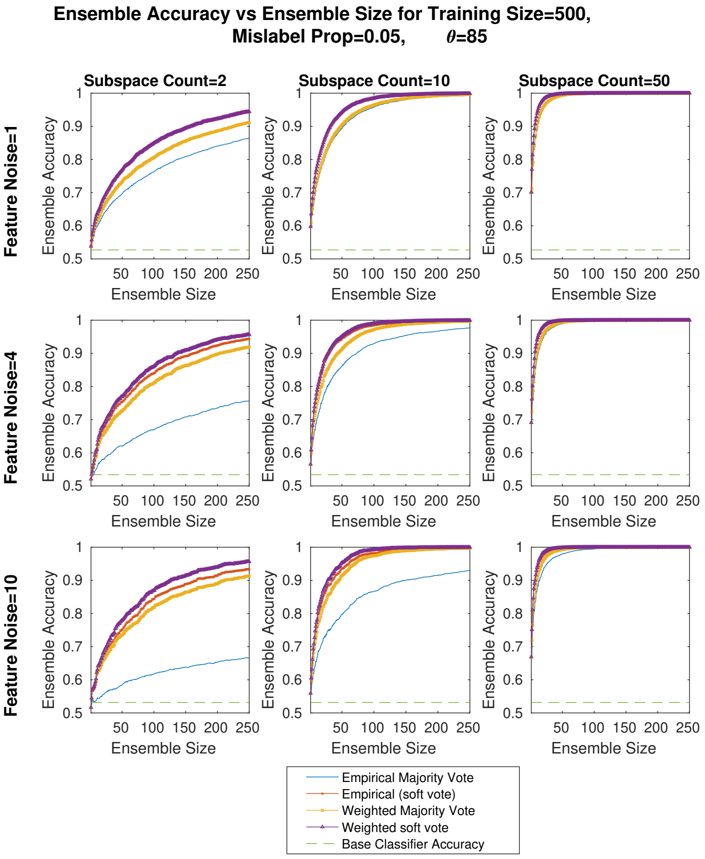

**Figure D.24:** *Ensemble classification accuracy vs ensemble member size for varying feature noise and projection dimensions, with training size $n = 500$, mislabelling proportion $q = 0.05$ and difficulty $\theta = 85$ and dimensionality $d = 1000$*

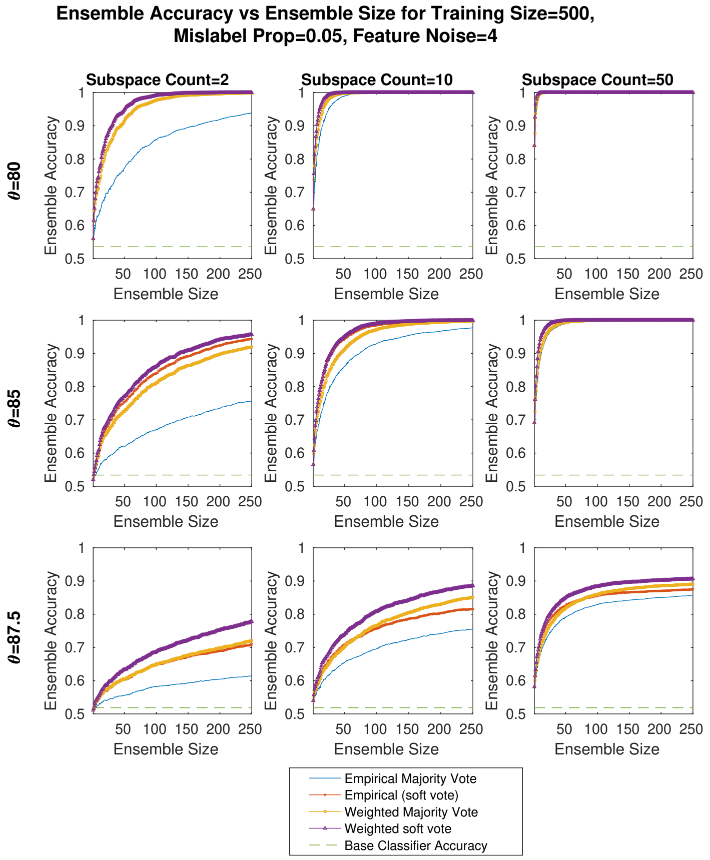

**Figure D.25:** *Ensemble classification accuracy vs ensemble member size for varying difficulty and projection dimensions, with training size $n = 500$, mislabelling proportion $q = 0.05$ and feature noise $s = 4$ and dimensionality $d = 1000$*

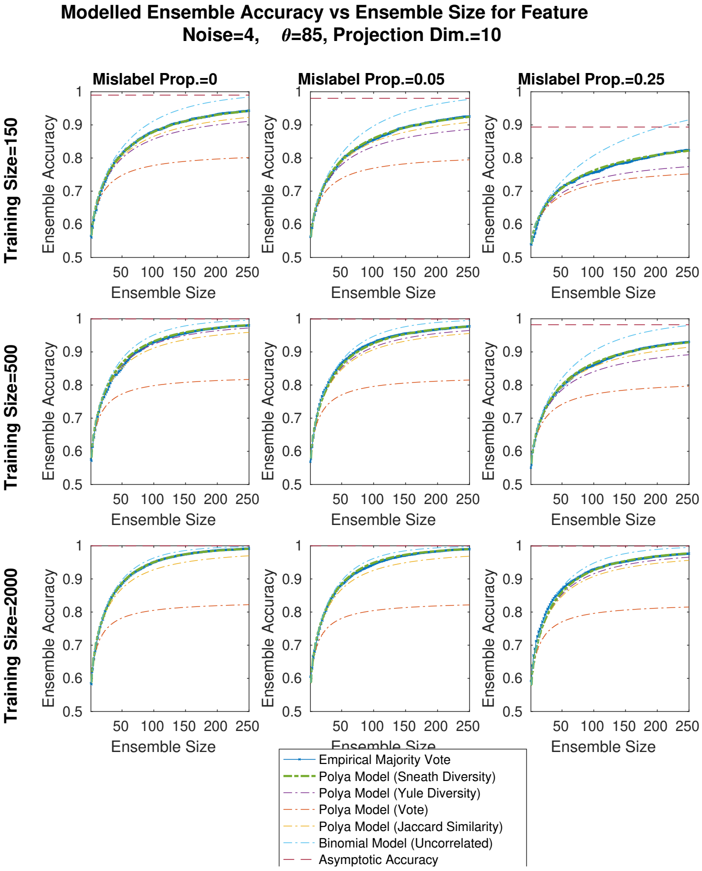

**Figure D.26:** *Majority vote ensemble accuracy as modelled by a Polya-Eggenberger distribution vs ensemble member size for varying training size and mislabelling proportion, with feature noise $s = 4$, difficulty $\theta = 85$ and projection dimensions $k = 10$ and dimensionality $d = 1000$*

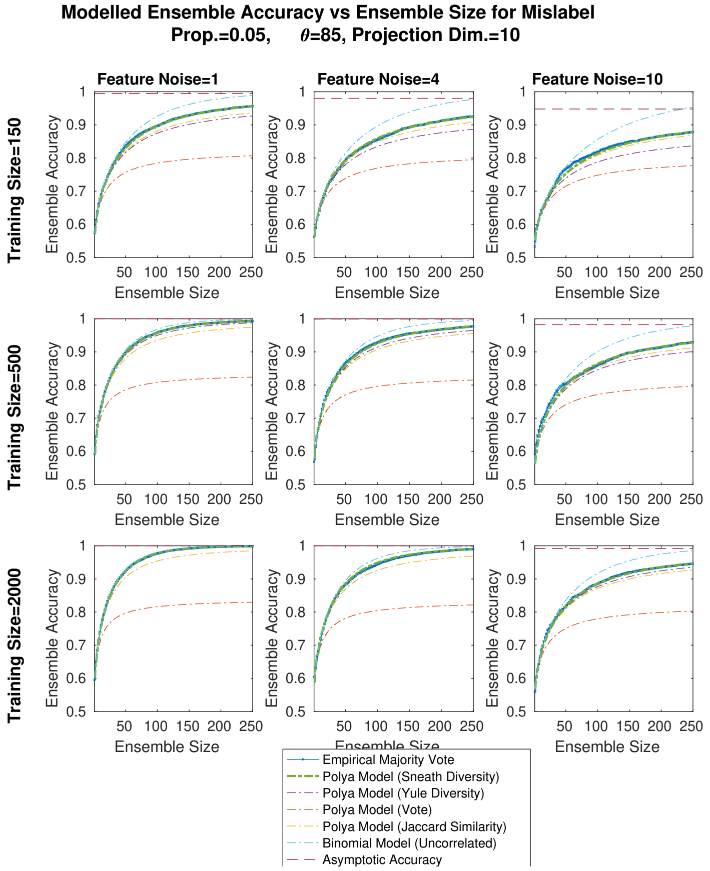

**Figure D.27:** *Majority vote ensemble accuracy as modelled by a Polya-Eggenberger distribution vs ensemble member size for varying training size and feature noise, with mislabelling proportion $q = 0.05$, difficulty $\theta = 85$ and projection dimensions $k = 10$ and dimensionality $d = 1000$*

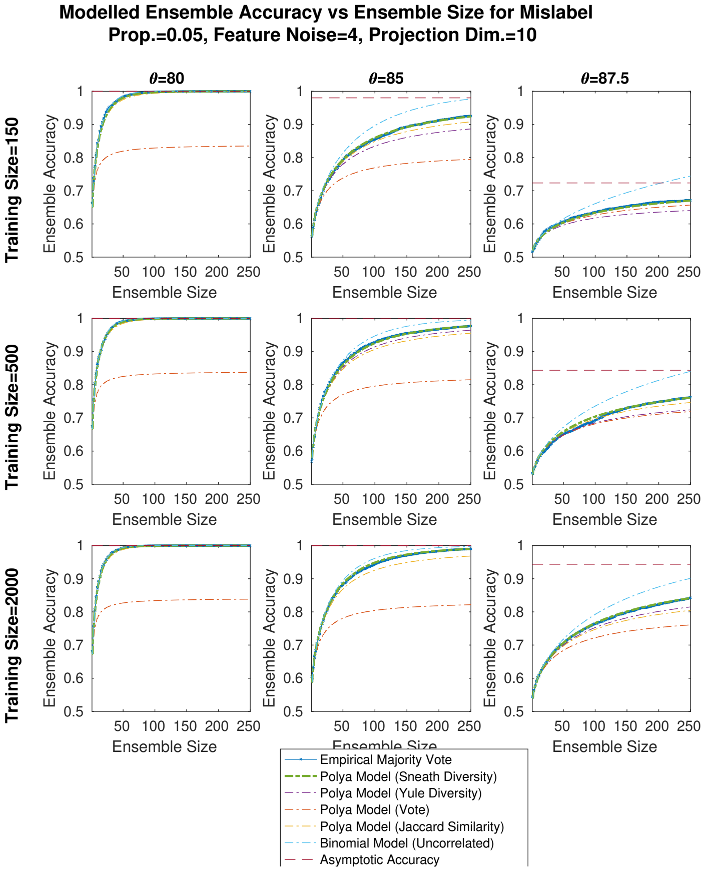

**Figure D.28:** *Majority vote ensemble accuracy as modelled by a Polya-Eggenberger distribution vs ensemble member size for varying training size and difficulty, with mislabelling proportion $q = 0.05$, feature noise $s = 4$ and projection dimensions $k = 10$ and dimensionality $d = 1000$*

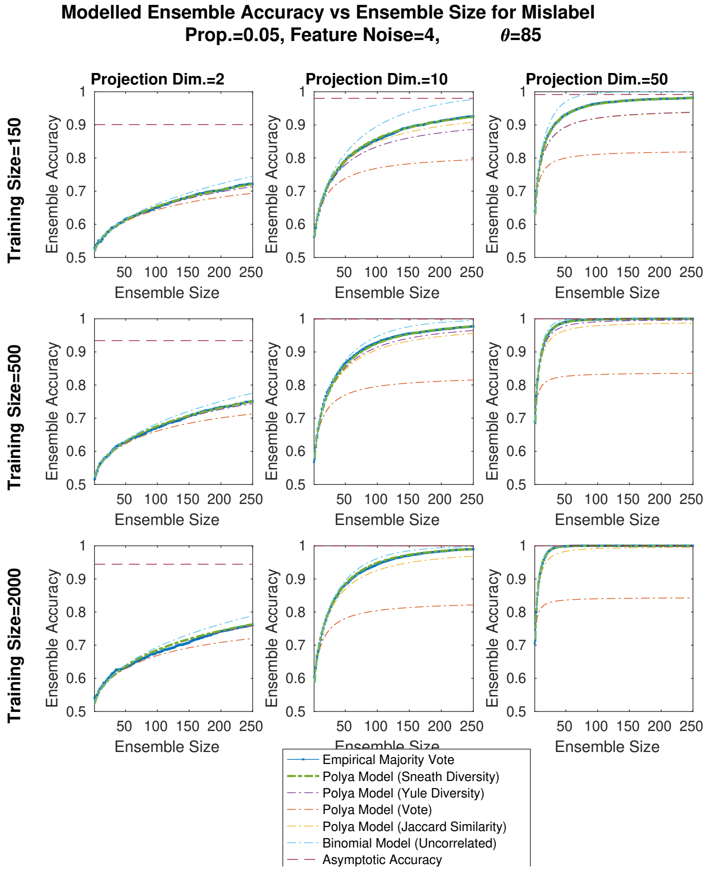

**Figure D.29:** *Majority vote ensemble accuracy as modelled by a Polya-Eggenberger distribution vs ensemble member size for varying training size and projection dimensions, with mislabelling proportion $q = 0.05$, feature noise $s = 4$ and difficulty $\theta = 85$ and dimensionality $d = 1000$*

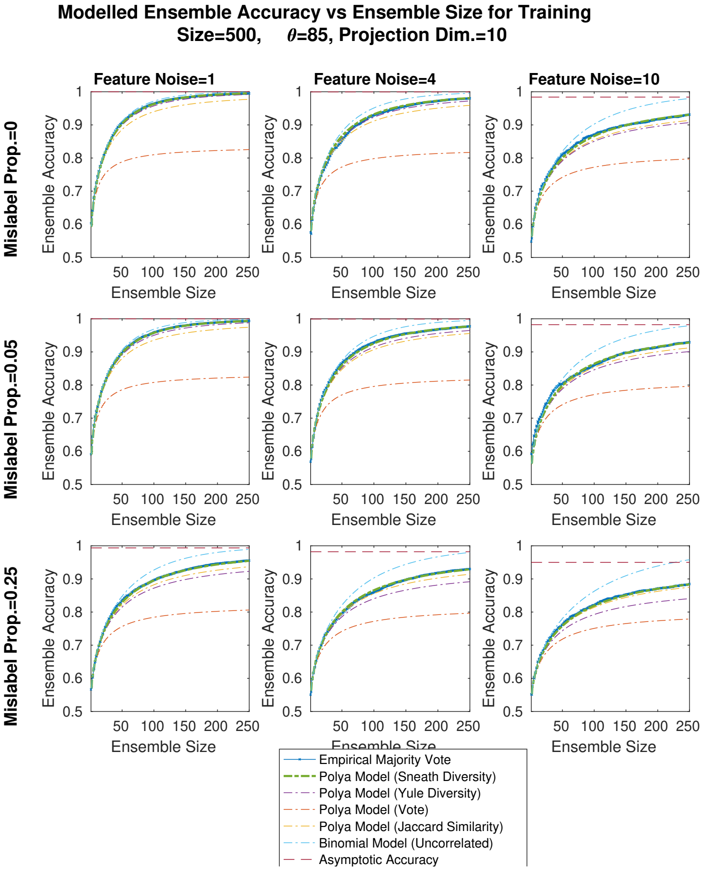

**Figure D.30:** *Majority vote ensemble accuracy as modelled by a Polya-Eggenberger distribution vs ensemble member size for varying mislabelling proportion and feature noise, with training size $n = 500$, difficulty $\theta = 85$ and projection dimensions $k = 10$ and dimensionality $d = 1000$*

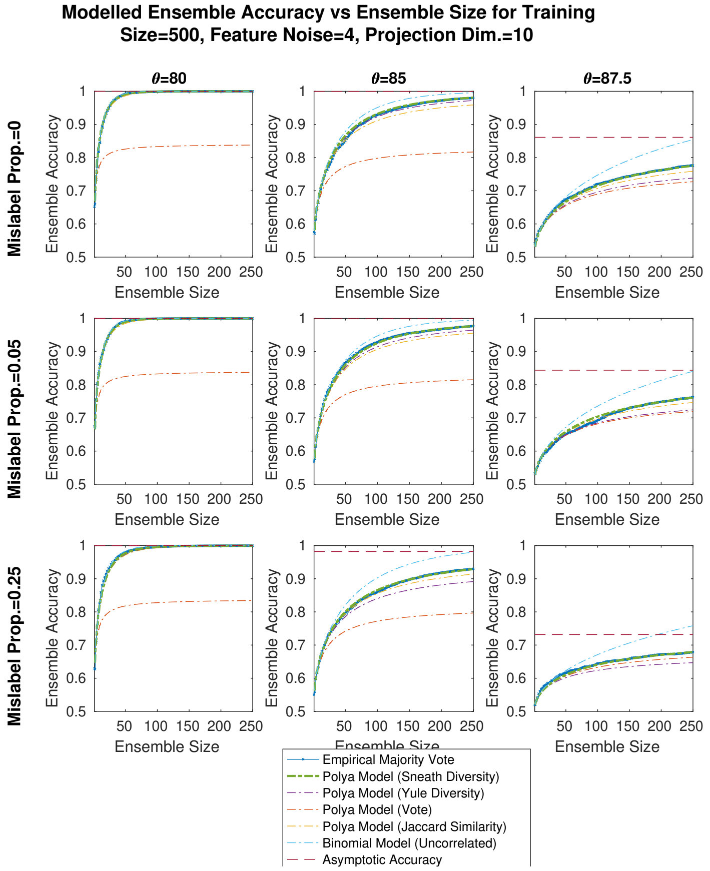

**Figure D.31:** *Majority vote ensemble accuracy as modelled by a Polya-Eggenberger distribution vs ensemble member size for varying mislabelling proportion and difficulty, with training size $n = 500$, feature noise $s = 4$ and projection dimensions $k = 10$ and dimensionality $d = 1000$*

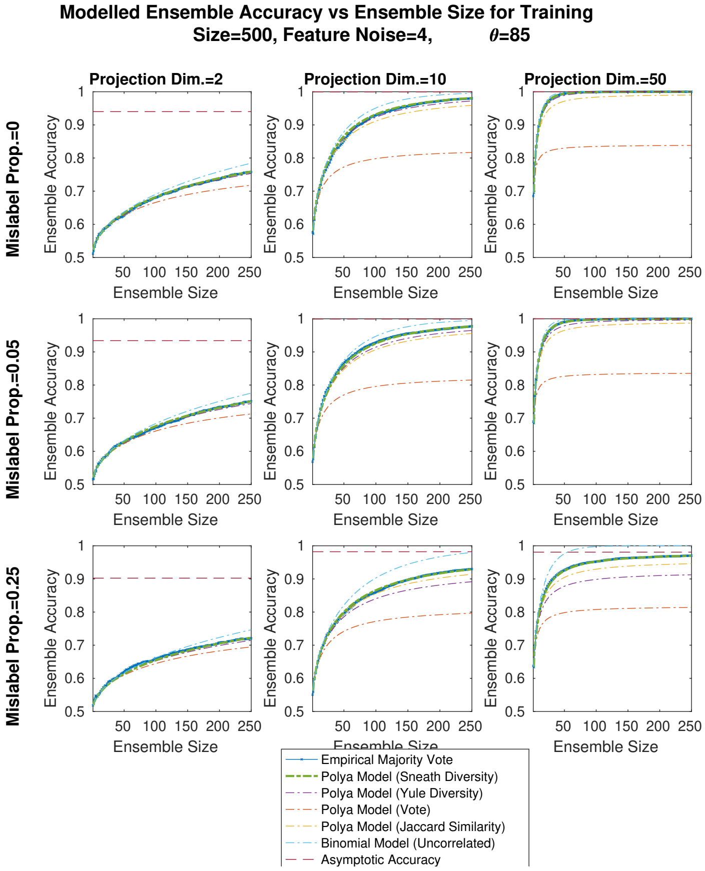

**Figure D.32:** *Majority vote ensemble accuracy as modelled by a Polya-Eggenberger distribution vs ensemble member size for varying mislabelling proportion and projection dimensions, with training size $n = 500$, feature noise $s = 4$ and difficulty $\theta = 85$ and dimensionality $d = 1000$*

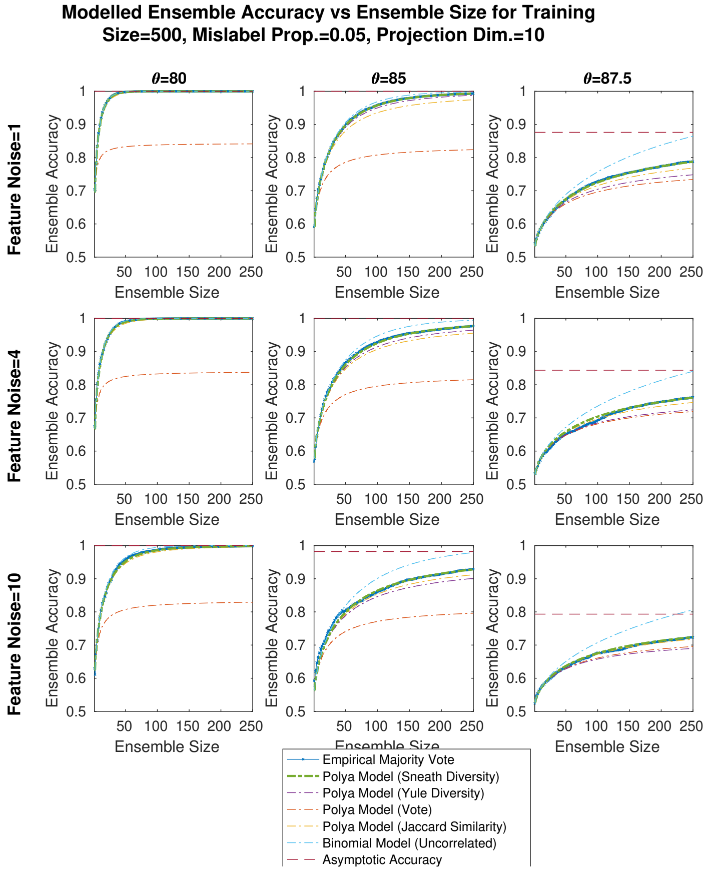

**Figure D.33:** *Majority vote ensemble accuracy as modelled by a Polya-Eggenberger distribution vs ensemble member size for varying feature noise and difficulty, with training size $n = 500$, mislabelling proportion $q = 0.05$ and projection dimensions $k = 10$ and dimensionality $d = 1000$*

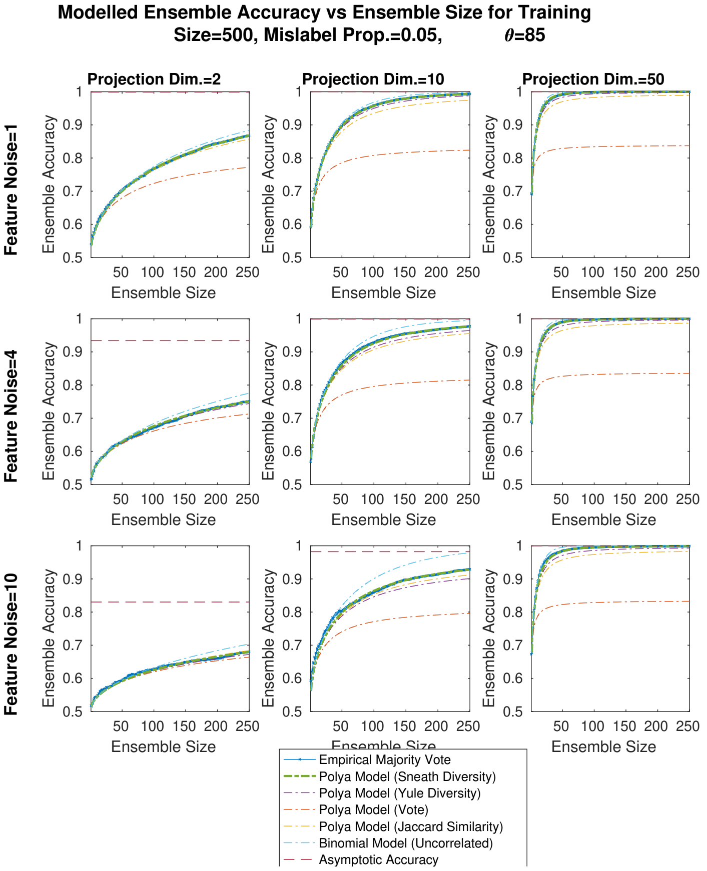

**Figure D.34:** *Majority vote ensemble accuracy as modelled by a Polya-Eggenberger distribution vs ensemble member size for varying feature noise and projection dimensions, with training size $n = 500$, mislabelling proportion $q = 0.05$ and difficulty $\theta = 85$ and dimensionality $d = 1000$*

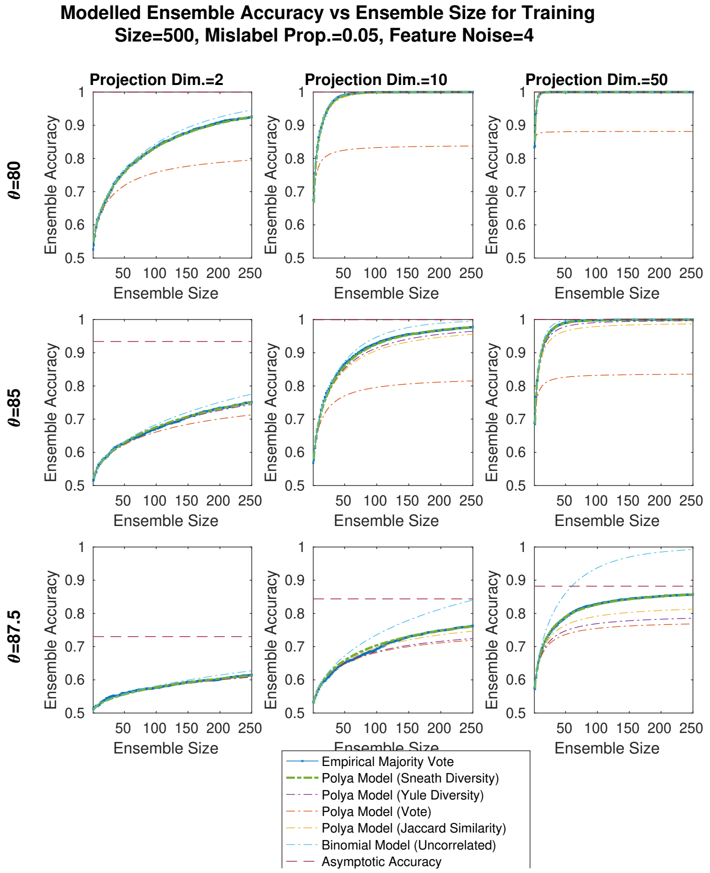

**Figure D.35:** *Majority vote ensemble accuracy as modelled by a Polya-Eggenberger distribution vs ensemble member size for varying difficulty and projection dimensions, with training size $n = 500$, mislabelling proportion $q = 0.05$ and feature noise $s = 4$ and dimensionality $d = 1000$*

## D.3 Appendix to Chapter 7

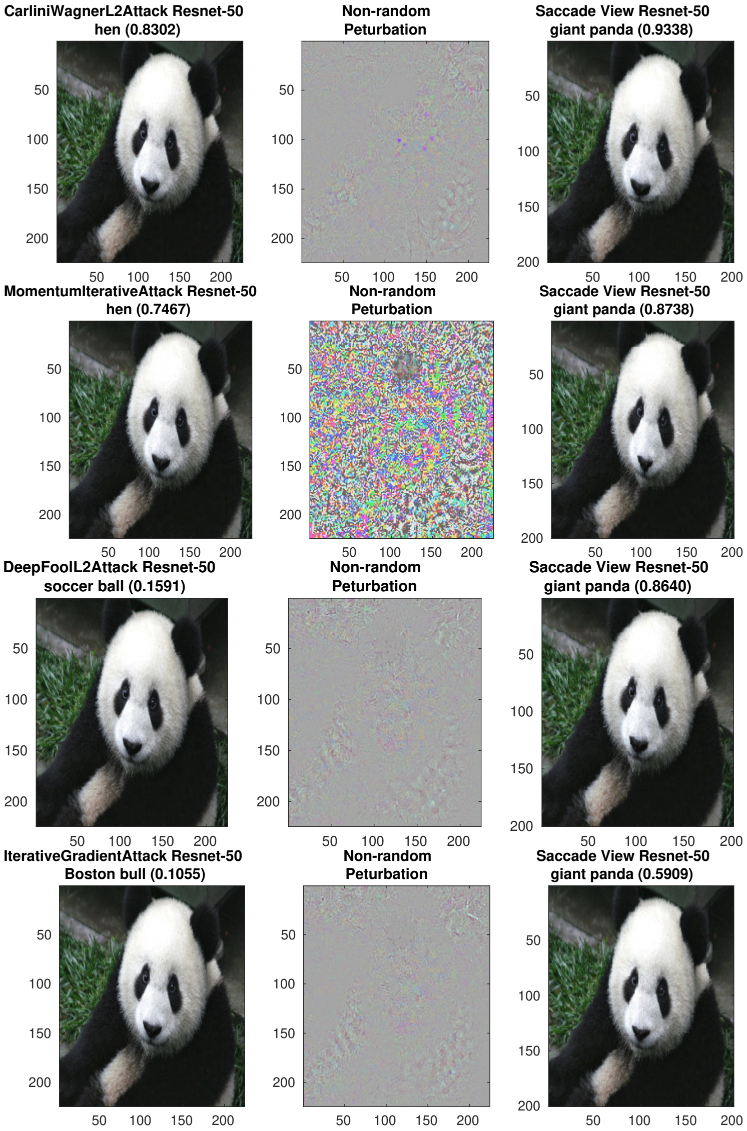

**Figure D.36:** *ResNet-50 classification of "gradient-based" adversarial attacks on the original form (left), and PseudoSaccade form (right). Image in the centre column is a visual representation of the adversarial attack. Observe that classification is very robust on the PseudoSaccades forms, returning a high confidence prediction on the true label.*

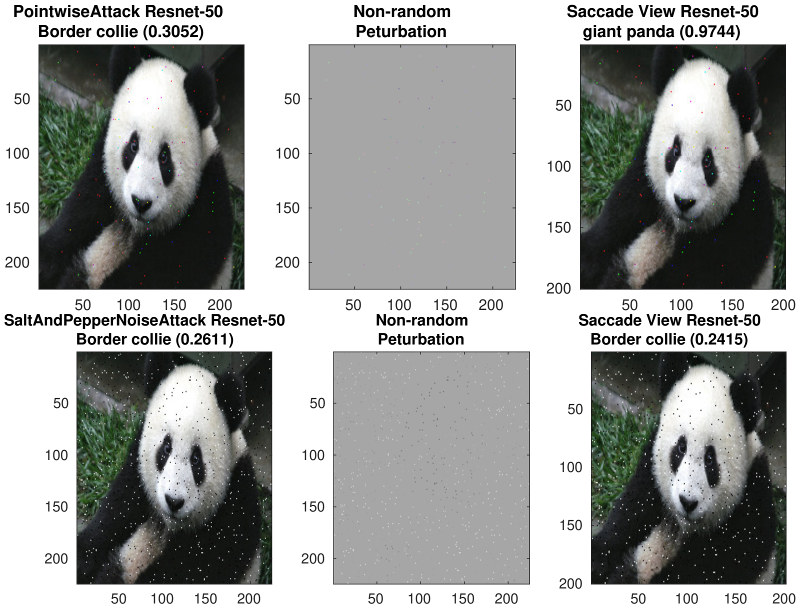

**Figure D.37:** *ResNet-50 classification of "pixel-based" adversarial attacks on the original form (left), and PseudoSaccade form (right). Image in the centre column is a visual representation of the adversarial attack. Observe that classification is somewhat robust on the PseudoSaccades forms, returning the moderate confidence predictions on the true label.*

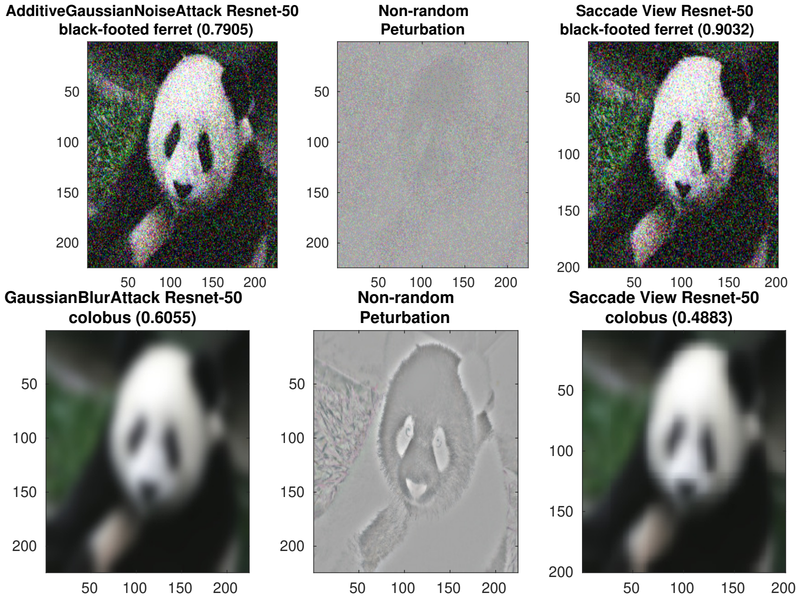

**Figure D.38:** *ResNet-50 classification of "contrast-based" adversarial attacks on the original form (left), and PseudoSaccade form (right). Image in the centre column is a visual representation of the adversarial attack. Observe that classification is not robust on the PseudoSaccades forms, returning an incorrect or low confidence predictions on the true label.*

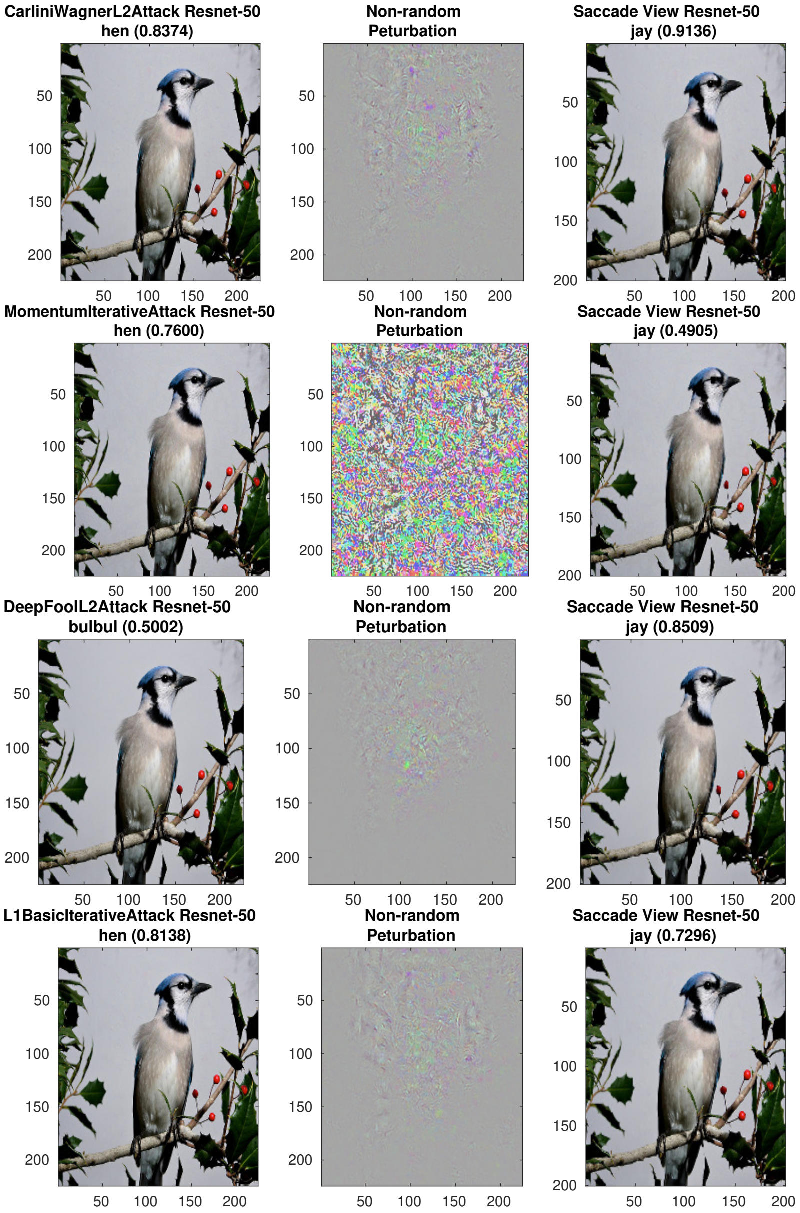

**Figure D.39:** *ResNet-50 classification of "gradient-based" adversarial attacks on the original form (left), and PseudoSaccade form (right). Image in the centre column is a visual representation of the adversarial attack. Observe that classification is very robust on the PseudoSaccades forms, returning a high confidence prediction on the true label.*

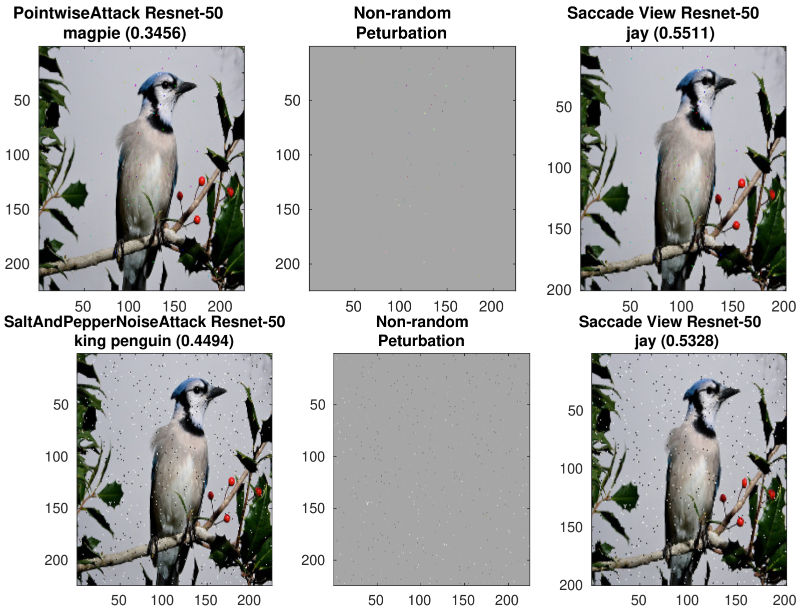

**Figure D.40:** *ResNet-50 classification of "pixel-based" adversarial attacks on the original form (left), and PseudoSaccade form (right). Image in the centre column is a visual representation of the adversarial attack. Observe that classification is somewhat robust on the PseudoSaccades forms, returning the moderate confidence predictions on the true label.*

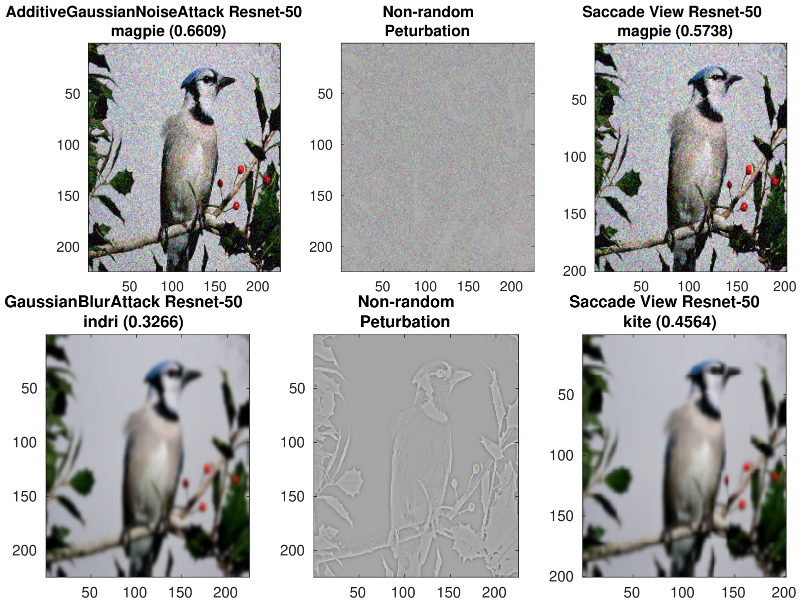

**Figure D.41:** *ResNet-50 classification of "contrast-based" adversarial attacks on the original form (left), and PseudoSaccade form (right). Image in the centre column is a visual representation of the adversarial attack. Observe that classification is not robust on the PseudoSaccades forms, returning an incorrect or low confidence predictions on the true label.*

| **True Label** | **Base Labels** | **Saccade Labels** | **Ensemble Labels** |
|---|---|---|---|
| **Cleaver** | **Cleaver (4)**<br>Carpenter's Kit (5) | **Cleaver (3)**<br>Carpenter's Kit (5) | **Cleaver (4)**<br>Carpenter's kit (5) |
| **Spatula** | **Spatula (3)** | **Spatula (2)** | **Spatula (2)** |
| **Sunscreen** | **Sunscreen (4)**<br>lotion (5) | **Sunscreen (3)**<br>lotion (5)<br>packet (3)<br>ice lolly (3) | **Sunscreen (3)**<br>lotion (5)<br>packet (3)<br>ice lolly (4) |
| **Tub** | **Tub (4)**<br>bathtub (14)<br>washbasin (4) | **Tub (4)**<br>bathtub (15)<br>washbasin (3) | **Tub (6)**<br>bathtub (12)<br>washbasin (3)\ |
| **Velvet** | **Velvet (4)**<br>purse (3)<br>wool (3) | **Velvet (4)**<br>purse (3)<br>wool (3) | **Velvet (5)**<br>wool (3) |
| **Projectile** | **Projectile (5)**<br>missile (15) | **Projectile (3)**<br>missile (16) | **Projectile (5)**<br>missile (16) |
| **Screwdriver** | **Screwdriver (6)** | **Screwdriver (4)**<br>padlock (3) | **Screwdriver (5)**<br>padlock (3) |
| **Hair Spray** | **Hair Spray (5)**<br>nipple (3)<br>water bottle (3) | **Hair Spray(6)**<br>lotion (3)<br>soap dispenser (3) | **Hair Spray (4)** |

**Table D.8:** *Labels where AlexNet Imagenet classifier achieved* $\leq 10\%$ *recall. Number of instances for which the given label was returned by classifier in brackets. Note that there are 50 instances for each of the classes. We omitted all predictions that occurred two or less times therefore the sum of the instances does not total to 50.*

| True Label | Base Label | Saccade Label | Ensemble Label |
|---|---|---|---|
| **Border terrier** | **Border terrier (21)**<br>bloodhound (3) | **Border terrier (24)** | **Border terrier (27)** |
| **Ram** | **Ram (16)**<br>bighorn sheep (16) | **Ram (20)**<br>bighorn sheep (10)<br>llama (3) | **Ram (23)**<br>bighorn sheep (12) |
| **bikini** | **bikini (17)**<br>maillot (7)<br>swimming trunks (3) | **bikini (20)**<br>maillot (3)<br>swimming trunks (3)<br>tub (3) | **bikini (22)**<br>maillot (3)<br>swimming trunks (3) |
| **cardigan** | **cardigan (23)**<br>suit (3) | **cardigan (27)**<br>stole (3) | **cardigan (29)**<br>stole (3) |
| **harvester** | **harvester (25)**<br>thresher (5)<br>tractor (6) | **harvester (32)**<br>thresher (3)<br>tractor (5) | **harvester (30)**<br>thresher (4)<br>tractor (5) |
| **lawn mower** | **croquet ball (3)**<br>go-kart (3)<br>lawn mower (29) | **lawn mower (29)**<br>croquet ball (3) | **lawn mower (34)**<br>croquet ball (3) |
| **mitten** | **mitten (28)**<br>Christmas stocking (3)<br>sock (3) | **mitten (21)**<br>Christmas stocking (3) | **mitten (33)**<br>Christmas stocking (3) |
| **prison** | **prison (22)**<br>shoji (3) | **prison (26)**<br>shoji (3) | **prison (27)**<br>shoji (3) |
| **shopping cart** | **shopping cart (24)** | **shopping cart (27)** | **shopping cart (31)** |
| **bell pepper** | **bell pepper (32)** | **bell pepper (36)** | **bell pepper (37)** |

**Table D.9:** *Labels where ensemble method performed significantly better* ($\geq 10\%$) *than the baseline AlexNet Imagenet classifier. Number of instances for which the given label was returned by classifier in brackets. Note that there are 50 instances for each of the classes. We omitted all predictions that occurred two or less times therefore the sum of the instances does not total to 50.*

| True Label | Base Label | Saccade Label | Ensemble Label |
|---|---|---|---|
| **bighorn sheep** | **bighorn sheep (30)**<br>ram (8) | **bighorn sheep (26)**<br>ram (12) | **bighorn sheep (25)**<br>ram (12) |
| **hamper** | **hamper (29)**<br>shopping basket (4) | **hamper (25)**<br>shopping basket (6) | **hamper (24)**<br>shopping basket (6) |

**Table D.10:** *Labels where ensemble method performed significantly worse* ($\geq 10\%$) *than the baseline AlexNet Imagenet classifier. Number of instances for which the given label was returned by classifier in brackets. Note that there are 50 instances for each of the classes. We omitted all predictions that occurred two or less times therefore the sum of the instances does not total to 50.*

| **True Label** | **Base Label** | **Saccade Label** | **Ensemble Label** |
|---|---|---|---|
| **CRT Screen** | **CRT screen (3)**<br>desk (7)<br>desktop computer (7)<br>monitor(12)<br>television (7) | **CRT screen (3)**<br>desk (6)<br>desktop computer (6)<br>monitor (13)<br>television (7) | **CRT screen (2)**<br>desk (7)<br>desktop computer (7)<br>monitor (13)<br>television (7) |
| **velvet** | **velvet (2)**<br>purse (3)<br>studio couch (3)<br>wool (3) | **velvet (2)**<br>cardigan (3) | **velvet (3)**<br>cardigan (3)<br>studio couch (3)<br>wool (3) |

**Table D.11:** *Labels where GoogLeNet Imagenet classifier achieved* $\leq 10\%$ *recall. Number of instances for which the given label was returned by classifier in brackets. Note that there are 50 instances for each of the classes. We omitted all predictions that occurred two or less times therefore the sum of the instances does not total to 50.*

| **True Label** | **Base Label** | Saccade Label | **Ensemble Label** |
|---|---|---|---|
| **water snake** | **water snake (20)**<br>ringneck snake (4) | **water snake (25)**<br>sea snake (3) | **water snake (26)**<br>ringneck snake (3) |
| **toy terrier** | **toy terrier (19)**<br>Chihuahua (6)<br>miniature pinscher (8)<br>basenji (5) | **toy terrier (22)**<br>Chihuahua (7)<br>miniature pinscher (8)<br>basenji (4) | **toy terrier (24)**<br>Chihuahua (6)<br>miniature pinscher (8)<br>basenji (3) |
| **wire-haired fox terrier** | **wire-haired fox terrier (25)**<br>Lakeland terrier (14) | **wire-haired fox terrier (28)**<br>Lakeland terrier (13) | **wire-haired fox terrier (30)**<br>Lakeland terrier (12) |
| **Bouvier des Flandres** | **Bouvier des Flandres (26)**<br>giant schnauzer (4) | **Bouvier des Flandres (29)**<br>giant schnauzer (5) | **Bouvier des Flandres (31)**<br>giant schnauzer (4) |
| **sea cucumber** | **sea cucumber (27)**<br>sea slug (3) | **sea cucumber (31)** | **sea cucumber (32)**<br>sea slug (3) |
| **mink** | **mink (28)** | **mink (33)** | **mink (33)** |
| **marmoset** | **marmoset (37)**<br>squirrel monkey (5) | **marmoset (39)**<br>squirrel monkey (4) | **marmoset (42)**<br>squirrel monkey (4) |
| **barbershop** | **barbershop (11)**<br>bakery (5)<br>barber chair (4)<br>restaurant (4)<br>tobacco shop (7) | **barbershop (15)**<br>bakery (3)<br>barber chair (3)<br>restaurant (4)<br>tobacco shop (7) | **barbershop (16)**<br>bakery (3)<br>barber chair (3)<br>restaurant (4)<br>tobacco shop (6) |
| **stone wall** | **stone wall (31)** | **stone wall (33)** | **stone wall (36)** |

**Table D.12:** *Labels where ensemble method performed significantly better* ($\geq 10\%$) *than the baseline GoogLeNet Imagenet classifier. Number of instances for which the given label was returned by classifier in brackets. Note that there are 50 instances for each of the classes. We omitted all predictions that occurred two or less times therefore the sum of the instances does not total to 50.*

| **True Label** | **Base Label** | **Saccade Label** | **Ensemble Label** |
|---|---|---|---|
| **polecat** | **polecat (24)**<br>weasel (4)<br>mink (4)<br>ferret (14) | **polecat (20)**<br>weasel (5)<br>mink (5)<br>ferret (15) | **polecat (19)**<br>weasel (4)<br>mink (6)<br>ferret (15) |
| **bathtub** | **bathtub (20)**<br>tub (14) | **bathtub (20)**<br>tub (13) | **bathtub (15)**<br>tub (15) |
| **bow** | **bow (29)** | **bow (26)** | **bow (24)** |
| **computer mouse** | **computer mouse (20)**<br>computer keyboard (3)<br>desktop computer (8) | **computer mouse (16)**<br>computer keyboard (4)<br>desktop computer (8) | **computer mouse (13)**<br>computer keyboard (3)<br>desktop computer (8)<br>notebook computer (3) |

**Table D.13:** *Labels where ensemble method performed significantly worse* ($\geq 10\%$) *than the baseline GoogLeNet Imagenet classifier. Number of instances for which the given label was returned by classifier in brackets. Note that there are 50 instances for each of the classes. We omitted all predictions that occurred two or less times therefore the sum of the instances does not total to 50.*

| True Label | Base Label | Saccade Label | Ensemble Label |
|---|---|---|---|
| **tiger cat** | **tiger cat (8)**<br>tabby cat (17)<br>Egyptian cat (5)<br>tiger (11) | **tiger cat (7)**<br>tabby cat (15)<br>Egyptian cat (7)<br>tiger (12) | **tiger cat (6)**<br>tabby cat (17)<br>Egyptian cat (4)<br>tiger (11) |
| **laptop computer** | **laptop computer (10)**<br>desktop computer (3)<br>notebook computer (29) | **laptop computer (13)**<br>notebook computer (27) | **laptop computer (13)**<br>desktop computer (3)<br>notebook computer (26) |
| **overskirt** | **overskirt (9)**<br>apron (3)<br>gown (3)<br>hoopskirt (6) | **overskirt (8)**<br>apron (3)<br>gown (4)<br>hoopskirt (6) | **overskirt (10)**<br>apron (3)<br>gown (3)<br>hoopskirt (6) |
| **CRT screen** | **CRT screen (6)**<br>desk (4)<br>desktop computer (7)<br>monitor (8)<br>television (11) | **CRT screen (5)**<br>desk (5)<br>desktop computer (7)<br>monitor (7)<br>television (11) | **CRT screen (5)**<br>desk (6)<br>desktop computer (8)<br>monitor (7)<br>television (10) |
| **sunglass** | **sunglass (10)**<br>seat belt (3)<br>sunglasses (15) | **sunglass (10)**<br>sunglasses (13) | **sunglass (11)**<br>sunglasses (11) |
| **velvet** | **velvet (9)**<br>purse (4) | **velvet (5)** | **velvet (8)**<br>quilt (3) |
| **Windsor tie** | **windsor tie (9)**<br>lab coat (3)<br>suit (11)<br>groom (4) | **windsor tie (9)**<br>lab coat (4)<br>suit (13)<br>groom (3) | **windsor tie (8)**<br>lab coat (4)<br>suit (12) |

**Table D.14:** *Labels where ResNet-50 Imagenet classifier achieved ≤ 20% recall. Number of instances for which the given label was returned by classifier in brackets. Note that there are 50 instances for each of the classes. We omitted all predictions that occurred two or less times therefore the sum of the instances does not total to 50.*

| **True Label** | **Base Label** | **Saccade Label** | **Ensemble Label** |
|---|---|---|---|
| **wolf spider** | **wolf spider (35)**<br>barn spider (4) | **wolf spider (38)** | wolf spider (40) |
| **German short-haired pointer** | **German short-haired pointer (36)**<br>Great Dane (3) | **German short-haired pointer (34)**<br>Hungarian Pointer (3) | **German short-haired pointer (41)** |
| **diaper** | diaper (29) | **diaper (32)** | **diaper (36)** |
| **tub** | **tub (12)**<br>bathtub (29) | **tub (15)**<br>bathtub (27) | **tub (17)**<br>bathtub (25) |
| **book jacket** | **book jacket (25)**<br>packet (4)<br>comic book (11) | **book jacket (28)**<br>packet (3)<br>comic book (8) | **book jacket (30)**<br>packet (5)<br>comic book (6) |

**Table D.15:** *Labels where ensemble method performed significantly better* (≥ 10%) *than the baseline ResNet-50 Imagenet classifier. Number of instances for which the given label was returned by classifier in brackets. Note that there are 50 instances for each of the classes. We omitted all predictions that occurred two or less times therefore the sum of the instances does not total to 50.*

| **True Label** | **Base Label** | Saccade Label | Ensemble Label |
|---|---|---|---|
| **electric ray** | **electric ray (35)**<br>stingray (4) | **electric ray (32)**<br>stingray (4) | **electric ray (30)**<br>stingray (5) |
| **Siberian husky** | **Siberian husky (31)**<br>Eskimo dog (9)<br>malamute (6) | **Siberian husky (27)**<br>Eskimo dog (12)<br>malamute (6) | **Siberian husky (25)**<br>Eskimo dog (12)<br>malamute (6) |
| **bannister** | **bannister (32)**<br>coil (4) | **bannister (25)**<br>coil (4)<br>prison (4) | **bannister (26)**<br>coil (4) |
| **hair spray** | **hair spray (23)**<br>lotion (3)<br>web site (3) | **hair spray (20)**<br>lotion (5)<br>web site (3) | **hair spray (18)**<br>lotion (4)<br>web site (3) |
| **joystick** | **joystick (36)** | **joystick (30)**<br>electrical switch (3) | **joystick (31)** |
| **magnetic compass** | **magnetic compass (25)**<br>analog clock (3)<br>barometer (7) | **magnetic compass (17)**<br>analog clock (3)<br>barometer (7)<br>buckle (3)<br>stopwatch (4) | **magnetic compass (18)**<br>analog clock (3)<br>barometer (7)<br>stopwatch (4) |
| **computer mouse** | **computer mouse (23)**<br>computer keyboard (3)<br>desktop computer (7)<br>monitor (3) | **computer mouse (16)**<br>computer keyboard (4)<br>desktop computer (8) | **computer mouse (16)**<br>desk (4)<br>desktop computer (9) |
| **Petri dish** | **Petri dish (30)** | **Petri dish (24)**<br>jellyfish (3) | **Petri dish (25)**<br>jellyfish (3) |
| **pitcher** | **pitcher (27)**<br>teapot (3)<br>water jug (5) | **pitcher (22)**<br>goblet (3)<br>teapot (5)<br>vase (3)<br>water jug (4) | **pitcher (22)**<br>goblet (3)<br>teapot (3)<br>vase (3)<br>water jug (3) |
| **spindle** | **spindle (45)** | **spindle (38)** | spindle (38)<br>maraca (3)<br>wool (3) |
| **stethoscope** | **stethoscope (30)**<br>lab coat (5) | **stethoscope (24)**<br>lab coat (5) | **stethoscope (25)**<br>lab coat (6) |

**Table D.16:** *Labels where ensemble method performed significantly worse* ($\geq$ 10%) *than the baseline ResNet-50 Imagenet classifier. Number of instances for which the given label was returned by classifier in brackets. Note that there are 50 instances for each of the classes. We omitted all predictions that occurred two or less times therefore the sum of the instances does not total to 50.*

# Bibliography

Achlioptas, D. (2001). Database-friendly random projections. In Buneman, P., editor, *Proceedings of the Twentieth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pages 274–281, Santa Barbara, CA. ACM.

Ahn, H., Moon, H., Fazzari, M. J., Lim, N., Chen, J. J., and Kodell, R. L. (2007). Classification by ensembles from random partitions of high-dimensional data. *Computational Statistics & Data Analysis*, 51(12):6166–6179.

Ailon, N. and Chazelle, B. (2009). The fast Johnson-Lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on Computing*, 39(1):302–322.

Ailon, N. and Liberty, E. (2009). Fast dimension reduction using Rademacher series on dual BCH codes. *Discrete & Computational Geometry*, 42(4):615.

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Van Esesn, B. C., Awwal, A. A. S., and Asari, V. K. (2018). The history began from AlexNet: a comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*.

Anderson, J., Belkin, M., Goyal, N., Rademacher, L., and Voss, J. (2014). The more, the merrier: The blessing of dimensionality for learning large gaussian mixtures. In Balcan, M., Feldman, V., and Szepesvári, C., editors, *Proceedings of The 27th Conference on Learning Theory, (COLT)*, volume 35 of *JMLR Workshop and Conference Proceedings*, pages 1135–1164, Barcelona, Spain. JMLR.org.

Anthony, M. and Bartlett, P. L. (2009). *Neural network learning: Theoretical foundations.* Cambridge University Press, Cambridge, UK.

Arriaga, R. I. and Vempala, S. (1999). An algorithmic theory of learning: Robust concepts and random projection. In *40th Annual Symposium on Foundations* *of Computer Science, (FOCS)*, pages 616–623, New York, NY. IEEE Computer Society.

Arrow, K. J. (1950). A difficulty in the concept of social welfare. *Journal of Political Economy*, 58(4):328–346.

Athalye, A., Engstrom, L., Ilyas, A., and Kwok, K. (2017). Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397*.

Balzano, L., Nowak, R., and Ellenberg, J. (2010). Compressed sensing illustration with audio. Retrieved from http://web.eecs.umich.edu/~girasole/csaudio/. [Online; accessed 17-Feb-2019].

Baraniuk, R., Davenport, M., DeVore, R., and Wakin, M. (2008). A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263.

Barber, D. (2012). *Bayesian reasoning and machine learning*. Cambridge University Press, Cambridge, UK.

Bardenet, R. and Maillard, O.-A. (2015). Concentration inequalities for sampling without replacement. *Bernoulli*, 21(3):1361–1385.

Bellman, R. (1970). *Methods of nonlinear analysis*, volume 61A. Academic Press, Cambridge, MA.

Bengio, Y. et al. (2009). Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, 2(1):1–127.

Bennett, G. (1962). Probability inequalities for the sum of independent random variables. *Journal of the American Statistical Association*, 57(297):33–45.

Berg, A., Deng, J., and Fei-Fei, L. (2010). Large scale visual recognition challenge (ILSVRC), 2010. Retrieved from: http://www.image-net.org/challenges/LSVRC. [Online; accessed 4-Aug-2018].

Berg, S. (1993). Condorcet's jury theorem, dependency among jurors. *Social Choice and Welfare*, 10(1):87–95.

Bingham, E. and Mannila, H. (2001). Random projection in dimensionality reduction: applications to image and text data. In Lee, D., Schkolnick, M., Provost, F. J., and Srikant, R., editors, *Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 245–250, San Francisco, CA. ACM.

Blum, A. (1997). Empirical support for winnow and weighted-majority algorithms: Results on a calendar scheduling domain. *Machine Learning*, 26(1):5–23.

Bootkrajang, J. and Kabán, A. (2013). Boosting in the presence of label noise. *arXiv preprint arXiv:1309.6818*.

Borda, J. (1781). Mémoire sur les élections au scrutin. *Histoire de l'Academie Royale des Sciences, Paris*, 1781:657–664.

Boucheron, S., Lugosi, G., and Massart, P. (2013). *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, Oxford, UK.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Brown, G. (2004). *Diversity in Neural Network Ensembles*. PhD thesis, University of Birmingham. Winner, British Computer Society Distinguished Dissertation Award.

Brown, G. (2010). Ensemble learning. In Sammut, C. and Webb, G. I., editors, *Encyclopedia of Machine Learning*, pages 312–320. Springer, Boston, MA.

Brown, G. and Kuncheva, L. I. (2010). "Good" and "bad" diversity in majority vote ensembles. In Gayar, N. E., Kittler, J., and Roli, F., editors, *Multiple Classifier Systems*, volume 5597, pages 124–133. Springer, Berlin, Germany.

Brown, G., Wyatt, J., Harris, R., and Yao, X. (2005). Diversity creation methods: a survey and categorisation. *Information Fusion*, 6(1):5–20.

Brown, G. and Wyatt, J. L. (2003). The use of the ambiguity decomposition in neural network ensemble learning methods. In Fawcett, T. and Mishra, N., editors, *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, volume 3, pages 67–74, Washington, DC. AAAI Press.

Candes, E. and Romberg, J. (2005). $\ell_1$-magic: Recovery of sparse signals via convex programming. Retrieved from http://www.acm.caltech.edu/l1magic/downloads/l1magic.pdf. [Online; accessed 24-Feb-2017].

Candes, E., Romberg, J., and Tao, T. (2004). Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *arXiv preprint arXiv:math/0409186*.

Candes, E. J. (2008). The restricted isometry property and its implications for compressed sensing. *Comptes rendus mathematique*, 346(9-10):589–592.

Cannings, T. I. and Samworth, R. J. (2017). Random-projection ensemble classification. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(4):959–1035.

Carlini, N. and Wagner, D. (2017). Adversarial examples are not easily detected: Bypassing ten detection methods. In Thuraisingham, B. M., Biggio, B., Freeman, D. M., Miller, B., and Sinha, A., editors, *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, (AISec@CCS)*, pages 3–14, Dallas, TX. ACM.

Carlini, N. and Wagner, D. (2018). Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 1–7, San Francisco, CA. IEEE Computer Society.

Condorcet, M. d. (1785). *Essay on the Application of Mathematics to the Theory of Decision-Making*. De L'Imprimerie Royale, Paris, France.

Cunningham, P. and Carney, J. (2000). Diversity versus quality in classification ensembles based on feature selection. In de Mántaras, R. L. and Plaza, E., editors, *11th European Conference on Machine Learning (ECML)*, volume 1810 of *Lecture Notes in Computer Science*, pages 109–116, Barcelona, Spain. Springer.

Dasgupta, S. and Gupta, A. (2003). An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65.

Didaci, L., Fumera, G., and Roli, F. (2013). Diversity in classifier ensembles: Fertile concept or dead end? In Zhou, Z.-H., Roli, F., and Kittler, J., editors, *Multiple Classifier Systems*, volume 11, pages 37–48. Springer, Nanjing, China.

Dietrich, F. (2008). The premises of Condorcet's jury theorem are not simultaneously justified. *Episteme*, 5:56–73.

Dietrich, F. and Spiekermann, K. (2013). Epistemic democracy with defensible premises. *Economics & Philosophy*, 29(1):87–120.

Dietterich, T. G. (2000a). Ensemble methods in machine learning. In Kittler, J. and Roli, F., editors, *Multiple Classifier Systems*, volume 1, pages 1–15. Springer, Cagliari, Italy.

Dietterich, T. G. (2000b). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157.

Domeniconi, C. and Yan, B. (2004). Nearest neighbor ensemble. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, volume 1, pages 228–231, Cambridge, UK. IEEE Computer Society.

Donoho, D. L. et al. (2000). High-dimensional data analysis: The curses and blessings of dimensionality. In *AMS Conference on Math Challenges of the 21st Century*, page 375, Los Angeles, CA. AMS.

Durrant, R. J. (2013). *Learning in high dimensions with projected linear discriminants*. PhD thesis, University of Birmingham.

Durrant, R. J. (2014). The Unreasonable Effectiveness of Random Projections in Computer Science. Retrieved from http://www.cs.waikato.ac.nz/~bobd/HDM_BobDurrant.pdf.

Durrant, R. J. and Kabán, A. (2010). Flip probabilities for random projections of $\theta$-separated vectors. Technical report, Technical Report CSR-10-10, School of Computer Science, University of Birmingham.

Durrant, R. J. and Kabán, A. (2013). Sharp Generalization Error Bounds for Randomly-projected Classifiers. In *Proceedings of the 30th International Conference on Machine Learning, (ICML)*, volume 28, pages 693–701, Atlanta, GA. JMLR.org.

Durrant, R. J. and Kabán, A. (2014). Random projections as regularizers: learning a linear discriminant from fewer observations than dimensions. *Machine Learning*, 99(2):257–286.

Dwinnell, W. (2010). LDA: Linear discriminant analysis. Retrieved from: urlhttps://www.mathworks.com/matlabcentral/fileexchange/29673-lda-linear-discriminant-analysis. [Online; accessed 4-Dec-2018].

Elsayed, G., Shankar, S., Cheung, B., Papernot, N., Kurakin, A., Goodfellow, I., and Sohl-Dickstein, J. (2018). Adversarial examples that fool both computer vision and time-limited humans. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31, (NIPS)*, pages 3910–3920, Montreal, Canada. Neural Information Processing Systems.

Fazzari, M. J. (2007). *Ensemble Methods for Classification with Applications to Genomics*. PhD thesis, The Graduate School, Stony Brook University: Stony Brook, NY.

Feller, W. (2008). *An introduction to probability theory and its applications*, volume 2. John Wiley & Sons, Hoboken, NJ.

Fonseca, E., Pons Puig, J., Favory, X., Font Corbera, F., Bogdanov, D., Ferraro, A., Oramas, S., Porter, A., and Serra, X. (2017). Freesound datasets: a platform for the creation of open audio datasets. In Hu, X., Cunningham, S. J., Turnbull, D., and Duan, Z., editors, *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, pages 486–493, Suzhou, China. International Society for Music Information Retrieval.

Freund, Y. and Schapire, R. E. (1999). Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296.

Fürnkranz, J. (2002). Round robin classification. *Journal of Machine Learning Research*, 2:721–747.

Galton, F. (1907). Vox populi (the wisdom of crowds). *Nature*, 75:450–451.

Ghahramani, Z. (2003). Unsupervised learning. In Bousquet, O., von Luxburg, U., and Rätsch, G., editors, *Advanced Lectures on Machine Learning, ML Summer Schools*, volume 3176 of *Lecture Notes in Computer Science*, pages 72–112, Canberra, Australia. Springer.

Giacinto, G. and Roli, F. (2001). Design of effective neural network ensembles for image classification purposes. *Image and Vision Computing*, 19(9-10):699–707.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press, Cambridge, MA.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014a). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in neural information processing systems 27, (NIPS)*, pages 2672–2680, Montreal, Canada. Neural Information Processing Systems.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014b). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Gorban, A. N., Tyukin, I. Y., and Romanenko, I. (2016). The blessing of dimensionality: Separation theorems in the thermodynamic limit. *IFAC-PapersOnLine*, 49(24):64–69. 2th IFAC Workshop on Thermodynamic Foundations for a Mathematical Systems Theory TFMST 2016.

Gu, S. and Rigazio, L. (2014). Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*.

Gurbaxani, R. and Mishra, S. (2018). Traits & transferability of adversarial examples against instance segmentation & object detection. *arXiv preprint arXiv:1808.01452*.

Guyon, I. (2003). Design of experiments of the NIPS 2003 variable selection benchmark. Retrieved from http://clopinet.com/isabelle/Projects/NIPS2003/Slides/NIPS2003-Datasets.pdf. [Online; accessed 1-Jan-2019].

Guyon, I., Gunn, S., Ben-Hur, A., and Dror, G. (2004). Result analysis of the nips 2003 feature selection challenge. In Thrun, S., Saul, L. K., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16, (NIPS)*, pages 545–552, Vancouver and Whistler, Canada. MIT Press.

Hansen, L. K. and Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 12(10):993–1001.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 770–778, Las Vegas, NV. IEEE Computer Society.

Ho, T. K. (1995). Random decision forest. In *Third International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 278–282, Montreal, Canada. IEEE Computer Society.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 20(8):832–844.

Ho, T. K., Hull, J. J., and Srihari, S. N. (1994). Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 16(1):66–75.

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

Indyk, P. (2001). Algorithmic applications of low-distortion geometric embeddings. In *42nd Annual Symposium on Foundations of Computer Science, (FOCS)*, pages 10–33, Las Vegas, NV. IEEE Computer Society.

Indyk, P. and Motwani, R. (1998). Approximate nearest neighbors: towards removing the curse of dimensionality. In Vitter, J. S., editor, *Proceedings of the Thirtieth*

*Annual ACM Symposium on the Theory of Computing*, pages 604–613, Dallas, TX. ACM.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In Nedellec, C. and Rouveirol, C., editors, *10th European Conference on Machine Learning (ECML)*, volume 1398, pages 137–142, Chemnitz, Germany. Springer.

Johnson, N. L. and Kotz, S. (1977). *Urn models and their application; an approach to modern discrete probability theory.* Wiley, New York.

Kabán, A. (2015). Improved bounds on the dot product under random projection and random sign projection. In Cao, L., Zhang, C., Joachims, T., Webb, G. I., Margineantu, D. D., and Williams, G., editors, *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,*, pages 487–496, Sydney, Australia. ACM.

Kabán, A. and Durrant, R. J. (2017). Structure-aware error bounds for linear classification with the zero-one loss. *arXiv preprint arXiv:1709.09782.*

Kane, D. M. and Nelson, J. (2014). Sparser Johnson-Lindenstrauss transforms. *Journal of the ACM (JACM)*, 61(1):4.

Karotkin, D. and Paroush, J. (2003). Optimum committee size: Quality-versus-quantity dilemma. *Social Choice and Welfare*, 20(3):429–441.

Kohavi, R. and Wolpert, D. H. (1996). Bias plus variance decomposition for zero-one loss functions. In Saitta, L., editor, *Proceedings of the Thirteenth International Conference on Machine Learning (ICML*, volume 96, pages 275–83, Bari, Italy. Morgan Kaufmann.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Bartlett, P. L., Pereira, F. C. N., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25, (NIPS)*, pages 1097–1105, Lake Tahoe, NV. Neural Information Processing Systems.

Krogh, A. and Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. In Tesauro, G., Touretzky, D. S., and Leen, T. K., editors, *Advances in Neural Information Processing Systems 7, (NIPS)*, pages 231–238, Denver, CO. MIT Press.

Kucheryavskiy, S. (2018). Blessing of randomness against the curse of dimensionality. *Journal of Chemometrics*, 32(1):e2966.

Kuncheva, L. I. (2002). A theoretical study on six classifier fusion strategies. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 24(2):281–286.

Kuncheva, L. I. and Rodríguez, J. J. (2014). A weighted voting framework for classifiers ensembles. *Knowledge and Information Systems*, 38(2):259–275.

Kuncheva, L. I., Rodríguez, J. J., Plumpton, C. O., Linden, D. E. J., and Johnston, S. J. (2010). Random subspace ensembles for fMRI classification. *IEEE Transactions on Medical Imaging*, 29(2):531–542.

Kuncheva, L. I. and Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207.

Kuncheva, L. I., Whitaker, C. J., Shipp, C. A., and Duin, R. P. (2003). Limits on the majority vote accuracy in classifier fusion. *Pattern Analysis & Applications*, 6(1):22–31.

Kuncheva, L. I., Whitaker, C. J., Shipp, C. A., and Duin, R. P. W. (2000). Is independence good for combining classifiers? In *15th International Conference on Pattern Recognition, (ICPR)*, volume 2, pages 168–171, Barcelona, Spain. IEEE Computer Society.

Kurakin, A., Goodfellow, I., and Bengio, S. (2016). Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

Ladha, K. K. (1993). Condorcet's jury theorem in light of de Finetti's theorem. *Social Choice and Welfare*, 10(1):69–85.

Ladha, K. K. (1995). Information pooling through majority-rule voting: Condorcet's jury theorem with correlated votes. *Journal of Economic Behavior & Organization*, 26(3):353–372.

Lai, C., Reinders, M. J. T., and Wessels, L. (2006). Random subspace method for multivariate feature selection. *Pattern Recognition Letters*, 27(10):1067–1076.

Lam, L. and Suen, S. (1997). Application of majority voting to pattern recognition: an analysis of its behavior and performance. *IEEE Transactions on Systems, Man, and Cybernetics*, 27(5):553–568.

Larsen, K. G. and Nelson, J. (2014). The Johnson-Lindenstrauss lemma is optimal for linear dimensionality reduction. *arXiv preprint arXiv:1411.2404*.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436.

Leon, F., Floria, S.-A., and Bădică, C. (2017). Evaluating the effect of voting methods on ensemble-based classification. In Jedrzejowicz, P., Yildirim, T., and Czarnowski, I., editors, *IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pages 1–6, Gdynia, Poland. IEEE.

Leung, K. T. and Parker, D. S. (2003). Empirical comparisons of various voting methods in bagging. In Getoor, L., Senator, T. E., Domingos, P. M., and Faloutsos, C., editors, *ACM SIGKDD international conference on Knowledge discovery and data mining*, volume 9, pages 595–600, Washington, DC. ACM.

Li, X. and Zhao, H. (2009). Weighted random subspace method for high dimensional data classification. *Statistics and its Interface*, 2(2):153.

Liberty, E. and Zucker, S. W. (2009). The Mailman algorithm: A note on matrix–vector multiplication. *Information Processing Letters*, 109(3):179–182.

Lin, W.-Y., Liu, S., Lai, J.-H., and Matsushita, Y. (2018). Dimensionality's blessing: Clustering images by underlying distribution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5784–5793, Salt Lake City, UT. IEEE Computer Society.

Liu, G., Liu, Q., and Li, P. (2017). Blessing of dimensionality: Recovering mixture data via dictionary pursuit. *IEEE transactions on Pattern Analysis & Machine Intelligence*, 39(1):47–60.

Loupes, G. (2014). *Understanding Random Forest: From Theory to Practice*. PhD thesis, University of Liége.

MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge University Press, Cambridge, UK.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Malmasi, S. and Dras, M. (2015). Language identification using classifier ensembles. In Nakov, P., Zampieri, M., Osenova, P., Tan, L., Vertan, C., Ljubešić, N., and Tiedemann, J., editors, *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects*, pages 35–43, Hissar, Bulgaria. Association for Computational Linguistics.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., and Byers, A. H. (2011). *Big Data: The Next Frontier for Innovation, Competition, and Productivity*. McKinsey Global Institute, New York, NY.

Matoušek, J. (2008). On variants of the Johnson–Lindenstrauss lemma. *Random Structures & Algorithms*, 33(2):142–156.

Meng, X. (2013). Scalable simple random sampling and stratified sampling. In *Proceedings of the 30th International Conference on Machine Learning, (ICML)*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 531–539, Atlanta, GA. JMLR.org.

Michie, D., Spiegelhalter, D. J., and Taylor, C. (1994). *Machine learning: Neural and Statistical Classification*. Ellis Horwood, Upper Saddle River, NJ.

Minsky, M. and Papert, S. (1969). *Perceptrons: An introduction to computational geometry*. MIT Press, Cambridge, MA.

Moler, C. (2016). Compare Gram-Schmidt and Householder Orthogonalization Algorithms. Retrieved from: https://blogs.mathworks.com/cleve/2016/07/25/compare-gram-schmidt-and-householder-o [Online; accessed 12-Mar-2018].

Nogueira, S. and Brown, G. (2015). Measuring the stability of feature selection with applications to ensemble methods. In Schwenker, F., Roli, F., and Kittler, J., editors, *International Workshop on Multiple Classifier Systems*, volume 9132 of *Lecture Notes in Computer Science*, pages 135–146, Günzburg, Germany. Springer.

Oinas-Kukkonen, H. (2008). *Network analysis and crowds of people as sources of new organisational knowledge*. Informing Science Press, Santa Rosa, CA.

Paroush, J. (1997). Stay away from fair coins: A Condorcet jury theorem. *Social Choice and Welfare*, 15(1):15–20.

Partridge, D. and Krzanowski, W. (1997). Software diversity: practical statistics for its measurement and exploitation. *Information and software technology*, 39(10):707–717.

Pereda, E., García-Torres, M., Melián-Batista, B., Mañas, S., Méndez, L., and González, J. J. (2018). The blessing of dimensionality: Feature selection outperforms functional connectivity-based feature transformation to classify ADHD subjects from EEG patterns of phase synchronisation. *PloS one*, 13(8):e0201660.

Piao, Y., Piao, M., Jin, C. H., Shon, H. S., Chung, J.-M., Hwang, B., and Ryu, K. H. (2015). A new ensemble method with feature space partitioning for high-dimensional data classification. *Mathematical Problems in Engineering*, 2:1–12.

Rauber, J., Brendel, W., and Bethge, M. (2017). Foolbox: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131*.

Riesen, K. and Bunke, H. (2007). Classifier ensembles for vector space embedding of graphs. In Haindl, M., Kittler, J., and Roli:, F., editors, *International Workshop on Multiple Classifier Systems*, volume 4472, pages 220–230, Prague, Czech Republic. Springer.

Schapire, R. E. (1990). The Strength of Weak Learnability. *Machine Learning*, 5(2):197–227.

Schapire, R. E. (2013). Explaining AdaBoost. In Schoölkopf, B., Luo, Z., and Vovk, V., editors, *Empirical inference*, chapter 5, pages 37–52. Springer, Berlin, Germany.

Schapire, R. E. and Freund, Y. (2012). *Boosting: Foundations and algorithms.* MIT press, Cambridge, MA.

Selfridge, O. G. (1958). *Pandemonium: A paradigm for learning*, volume 1. National Physical Laboratory, London, UK.

Sen, K. and Mishra, A. (1996). A generalised Polya-Eggenberger model generating various discrete probability distributions. *Sankhyā: The Indian Journal of Statistics, Series A*, 58:243–251.

Serfling, R. J. (1974). Probability inequalities for the sum in sampling without replacement. *The Annals of Statistics*, 2(1):39–48.

Serpen, G. and Pathical, S. (2009). Classification in high-dimensional feature spaces: Random subsample ensemble. In Wani, M. A., Kantardzic, M. M., Palade, V., Kurgan, L. A., and Qi, Y. A., editors, *International Conference on Machine Learning and Applications (ICMLA)*, pages 740–745, Miami Beach, FL. IEEE Computer Society.

Serre, T., Wolf, L., and Poggio, T. (2006). Object recognition with features inspired by visual cortex. Technical report, Massachusetts Inst of Tech Cambridge Dept of Brain and Cognitive Sciences.

Sewell, M. (2011). Ensemble Learning. Technical report, UCL Department of Computer Science.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press, Cambridge, UK.

Sitawarin, C., Bhagoji, A. N., Mosenia, A., Chiang, M., and Mittal, P. (2018). Darts: Deceiving autonomous cars with toxic signs. *arXiv preprint arXiv:1802.06430.*

Skurichina, M. and Duin, R. P. W. (2002). Bagging, boosting and the random subspace method for linear classifiers. *Pattern Analysis & Applications*, 5(2):121–135.

Sneath, P. H. and Sokal, R. R. (1963). Principles of numerical taxonomy. *Taxon*, 12(5):190–199.

Sorzano, C. O. S., Vargas, J., and Montano, A. P. (2014). A survey of dimensionality reduction techniques. *arXiv preprint arXiv:1403.2877*.

Spruyt, V. (2014). The Curse of Dimensionality in classification. Retrieved from: http://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/. [Online; accessed 14-Feb-2016].

Stewart, G. (1998). *Matrix Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Sun, T. and Zhou, Z.-H. (2018). Structural diversity for decision tree ensemble learning. *Frontiers of Computer Science*, 12(3):560–570.

Sutton, R. S. (1996). Generalization in reinforcement learning: Successful examples using sparse coarse coding. In Touretzky, D. S., Mozer, M., and Hasselmo, M. E., editors, *Advances in neural information processing systems 8, (NIPS)*, pages 1038–1044, Denver, CO. MIT Press.

Sutton, R. S., Barto, A. G., et al. (1998). *Introduction to reinforcement learning*, volume 135. MIT press, Cambridge, MA.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 1–9, Boston, MA. IEEE Computer Society.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Tao, D., Tang, X., Li, X., and Wu, X. (2006). Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE transactions on Pattern Analysis & Machine Intelligence*, 28(7):1088–1099.

Tropp, J. A. (2009). Column subset selection, matrix factorization, and eigenvalue optimization. In Mathieu, C., editor, *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, (SODA)*, pages 978–986, New York, NY. SIAM.

Tumer, K. and Ghosh, J. (1996). Theoretical foundations of linear and order statistics combiners for neural pattern classifiers. Technical report, IEEE Transacation on neural networks.

Ueda, N. and Nakano, R. (1996). Generalization error of ensemble estimators. In *IEEE International Conference on Neural Networks*, volume 1, pages 90–95, Washington, DC. IEEE.

Valentini, G. and Masulli, F. (2002). Ensembles of learning machines. In Marinaro, M. and Tagliaferri, R., editors, *Neural Nets, 13th Italian Workshop on Neural Nets, (WIRN)*, volume 2486 of *Lecture Notes in Computer Science*, pages 3–20, Viertri sul Mare, Italy. Springer.

Van Erp, M., Vuurpijl, L., and Schomaker, L. (2002). An overview and comparison of voting methods for pattern recognition. In *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, pages 195–200, Ontario, Canada. IEEE Computer Society.

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999.

Vedaldi, A. and Lenc, K. (2015). Matconvnet: Convolutional neural networks for matlab. In Zhou, X., Smeaton, A. F., Tian, Q., Bulterman, D. C. A., Shen, H. T., Mayer-Patel, K., and Yan, S., editors, *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, (MM)*, pages 689–692, Brisbane, Australia. ACM.

Venkatasubramanian, S. and Wang, Q. (2011). The Johnson-Lindenstrauss transform: an empirical study. In Müller-Hannemann, M. and Werneck, R. F. F., editors, *Proceedings of the Thirteenth Workshop on Algorithm Engineering and Experiments, ALENEX*, pages 164–173, San Francisco, CA. SIAM.

Vershynin, R. (2018). *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge University Press, Cambridge, UK.

Vitter, J. S. (1985). Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57.

Wang, W. and Zhou, Z.-H. (2017). Theoretical foundation of co-training and disagreement-based algorithms. *arXiv preprint arXiv:1708.04403*.

Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., and De Freitas, N. (2015). Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581*.

Weber, A. G. (2006). USC - Signal and Image Processsing Institute. Retrieved from http://sipi.usc.edu/database/. [Online; accessed 24-Jan-2018].

Weisstein, E. W. (2002). "generalized hypergeometric function." from mathworld–a wolfram web resource. Retrieved from: http://mathworld.wolfram.com/GeneralizedHypergeometricFunction.html.

Whalen, S. and Pandey, G. (2013). A comparative analysis of ensemble classifiers: Case studies in genomics. In Xiong, H., Karypis, G., Thuraisingham, B. M., Cook, D. J., and Wu, X., editors, *IEEE 13th International Conference on Data Mining*, pages 807–816, Dallas, TX. IEEE Computer Society.

Whitaker, C. and Kuncheva, L. (2003). Examining the relationship between majority vote accuracy and diversity in bagging and boosting. Technical report, School of Informatics, University of Wales, Bangor.

Yang, L. (2011). Classifiers selection for ensemble learning based on accuracy and diversity. *Procedia Engineering*, 15:4266–4270.

Yule, G. U. (1900). On the association of attributes in statistics: with illustrations from the material of the childhood society. *Philosophical Transactions of the Royal Society of London*, 194:257–319.

Zhou, Z.-H. (2012). *Ensemble methods: foundations and algorithms.* CRC Press, Boca Raton, FL.

Zhou, Z.-H. (2017). A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53.

Zhu, X. J. (2005). Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.