# Niceness Assumptions for Learning Algorithms

by

Shrinu Kushagra

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2014

## Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

## Abstract

Various machine learning algorithms like Neural Networks, Linear Regression, Feature Learning etc. are being employed successfully in a wide variety of applications like computer vision, speech recognition, bioinformatics etc. However, many of these learning algorithms have been shown to be NP-Hard. Furthermore some of these algorithms are even hard NP-Hard approximate. The intuition behind the success of these algorithms is that in practical applications the input data is not 'worst-case' and has certain 'nice' properties. In this thesis, we take steps towards bridging the apparent gap between what is predicted by theory and what is actually happening in practice. We consider two different niceness assumptions.

The notion of Metric Distortion is fairly common for dimensionality reduction techniques. The goal is to obtain reduction techniques such that the distortion is small for all pairs of points. We show via an example that Metric Distortion is not good at modeling dimensionality reduction techniques which would perform quite well in practice. We introduce *Retaining Distances*, a probabilistic notion for modeling dimensionality reduction techniques which preserve most of the inter-point distances. Retaining Distance can viewed as a relaxation of Metric Distortion. We prove that common techniques like PCA can be modeled by our notion.

Another niceness assumption inherent in many machine learning algorithms is that 'close points tend to have same labels'. A notion of *Probabilistic Lipschitzness* (PL) was introduced by Urner et al. [28] to capture this intuition. In this work, we propose a new definition of PL. We show that both these definitions are orthogonal to one another, in the sense that, one is not implied by (or a relaxation of) the other. We give sample complexity upper bounds for Nearest Neighbor under this new definition.

The crux of the thesis is combining the two notions to show that information (niceness) is preserved across dimensions. We prove that if we have PL in a higher dimension and any dimensionality-reduction technique retains distances then we have PL in reduced dimension as well. That is, a distance retaining reduction preserves PL. In other words, the niceness properties that existed in the original dimension also exist in reduced dimension space.

Towards the end, we validate both our notions experimentally. We show how our notion of retaining distance maybe employed in practice to capture the 'usefulness' of a reduction technique. We also perform experiments to show how the two notions of PL compare in practice.

## Acknowledgements

I would like to thank my supervisor Prof. Shai Ben-David for his guidance in the past one and a half years. Prof. Shai gave me freedom to choose my own interest at my own pace and was always helpful with his suggestions and guidance. The numerous discussions over which definitions to use, minutely examining all the proofs and pointing out even the smallest of mistakes were helpful to build a solid background in the field of Learning Theory and also helped clear some of conceptual errors. I would also like to thank him for extending this association beyond the Masters program and agreeing to become my PhD supervisor as well. I would also like to thank Prof. Dan Lizotte and Prof. Jesse Hoey for agreeing to read my Thesis.

I would like to thank my close friends, Sandeep, Rakesh and Rupinder for making the two year stay at Waterloo so memorable. Hangouts, road trips, campings, numerous insightful discussions and laughter sessions made sure that I never missed home and this place became a home away from home. I would also like to thank Clive Porter, Waterloo Squash Team Coach for first introducing me to the game and then having the confidence to induct me in the Varsity team.

Finally, I would like to thank my parents for being a pillar of constant strength, backing all my career decisions and supporting me through thick and thin.

## Dedication

To my grandparents for their unconditional love and good wishes.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

The rapid rise of the internet and computing resources has led to the availability of a large amount of data. Machine Learning techniques and algorithms aim to learn from the data and make predictions. Some of the learning algorithms like decision trees, linear regression, logistic regression, neural networks, nearest neighbors have been applied successfully in a vast variety of fields including genetics, finance, language processing, weather prediction, astronomy and many more.

However a lot of the learning algorithms mentioned above have been shown to be NP-hard. For example Venkatesan et. al [18] showed that "weak agnostic proper learning of half-spaces" is hard. Blum et al. [12] showed that training a neural network is NP-Complete. Similarly, Rivest et. al [21] showed that decision tree learning is NP-Complete.

Despite these negative theoretical results for most of the popular machine learning algorithms, they continue to employed in practice rather successfully. This success can be attributed to the fact that the data we encounter in real-world applications is seldom 'worst-case'. In most of the cases, the input data and the distribution have certain nice properties which are exploited by the learning algorithms. For example, intuitively we attribute the success of nearest neighbor algorithm to the fact that points which are close in feature space tend to have same labels. Similarly, linear predictors would be successful when the input data can be separated using half-spaces with relatively small error. Similar assumptions are inherent in almost all the machine learning algorithms.

In this thesis, we want to mathematically model these assumptions and show that learning is possible under these 'niceness' assumptions. We consider two niceness assumptions. One assumption which we call *Retaining Distance* is used to model dimensionality reduction techniques and the other called *Probabilistic Lipschitzness* is used model labeling functions.

## 1.1 Niceness assumptions in Dimensionality Reduction

Real world applications need to deal with data that have a large number of features (or high dimension). A high dimensional data poses many challenges. The running time of many common classification algorithms, regression algorithms depend linearly or exponentially on the dimension of the data [5, 32]. As the dimension increases, the amount of data needed to make a prediction also grows exponentially. These problems associated with working with high dimensional data have been termed as the curse of dimensionality [10, 9].

Due to the prevalence of high dimensional data, dimensionality reduction techniques are used in many domains of Machine Learning as a preprocessing step. Some of the popular dimensionality reduction techniques include Principal Component Analysis (PCA), Laplacian Eigenmaps, Multidimensional Scaling, Isomap, Neural Autoencoders, K-means based Dimensionality Reduction and many more. A nice overview of these techniques can be found in [30, 15].

Dimensionality reduction techniques (sometimes also referred to as embeddings) aim to transform data from a high dimension space into a low dimension space while preserving the important properties/features of the data. One of such properties is inter-point distances. A lot of literature exists on finding embeddings which preserve inter-point distances for all pairs of points (details in Chapter 2). However, often a more realistic assumption is to enforce that the distance be preserved for most but not all the pairs of points. We show by constructing simple examples that this assumption is indeed more realistic and intuitive. We model this assumption by our notion of *Retaining Distance*.

PCA is perhaps the most popular dimensionality reduction technique used. In practical applications it is seen that if PCA is able to capture the variance of the data then it performs quite well. The better it captures the variance of the data the better is the performance of the reduced representations. We show that under exactly these assumptions, PCA can be modeled by our notion of Retaining Distance.

## 1.2 Niceness Assumptions for labeling functions

As discussed earlier, various classification algorithms rely on the assumption that the data labeling function has certain 'nice' properties. An assumption inherent in learning paradigms like nearest neighbor, linear classification is that close points tend to have similar labels. To model this assumption, the notion of *Probabilistic Lipschitzness* (PL) was introduced by Urner et al. [29].

The function which labels the data can be either deterministic or non-deterministic. For the deterministic case, the labeling function is well-defined. For the non-deterministic case, the probability over the labels can be used as the 'labeling' function (this is slight abuse of formal notation for simplicity). In this thesis, we restrict ourselves to the case when the labeling function is deterministic.

In Mathematics, continuity is a notion which is very commonly used for functions. Lipschitzness is a stronger form of continuity. Theoretical results (sample complexity upper bounds) are known when the conditional probability function over the labels is Lipschitz (Theorem 19.3 in [26]). Similar results were obtained by Urner et al. [28] when the labeling function is deterministic and satisfies PL. The notion of PL can be viewed as a relaxation of the standard Lipschitz condition. PL essentially says that the probability of generating a point such that the standard Lipschitz condition is violated is small and bounded. In this thesis, we introduce a different definition of PL and discuss its merits and demerits over the original definition both theoretically and experimentally.

## 1.3 Preserving niceness while reducing dimension

Consider the following scenario. In some high dimension space, close points tend to have same labels. We have a dimensionality reduction technique which preserves most of the inter-point distances. We use this reduction technique to reduce the dimension of our original data. Since most of the close points had same labels originally and our reduction preserved most of the inter-point distances, intuitively one would expect that most of the close points would have same labels even in the reduced dimension. In this thesis, we build a theoretical framework and formally prove the above intuition with the appropriate mathematical rigor. To summarize our result in a single sentence, "*A nice dimensionality reduction technique is such that if a labeling function was nice in original dimension then the labeling function remains nice in the reduced dimension*".

## 1.4 Outline of the thesis

The thesis is organized as follows. Chapter 2 presents all the related work in the area and then discusses the contributions that we have made. In Chapter 3 we introduce our notion of retaining distance and prove that PCA retains distance. Chapter 4 introduces our notion of Probabilistic Lipschitzness and gives sample complexity bounds for Nearest Neighbor under PL assumptions. Chapter 5 combines PL and Retaining Distance to show how retaining distances leads to niceness properties being preserved. Chapter 6 gives some applications of how our notion can be useful in practice. In Chapter 7, we present our experimental results. Chapter 8 concludes our work and discusses avenues for possible future work.

# Chapter 2

# Related Work

In this thesis, we introduce two assumptions to model the 'niceness' properties of distributions. We present some of the other notions prevalent in literature and discuss how our results are different.

## 2.1 Metric Distortion

Dimensionality reduction techniques (or embeddings) reduce the dimensionality of the data while preserving important properties (like inter-point distances) of the data. A notion of *distortion* is used to measure the fraction by which the distances have changed between a pair of points as the dimensionality of the space reduces. More formally,

**Definition 2.1 (Distortion).** Let $\mathbb{DR}$ be any dimensionality reduction technique (embedding) that takes points in $\mathbb{R}^N$ and returns points in $\mathbb{R}^n$. Let $d$ and $d'$ be distance functions in the two spaces respectively. Let $d''$ be the extension of $d'$ to $\mathbb{R}^N$. Then distortion between a pair of points $x, y$ is defined as

$$dist(x,y) = \frac{d(x,y)}{d''(x,y)}.\text{ And distortion of the embedding } \mathbb{DR} \text{ is defined as}$$

$$dist(\mathbb{DR}) = \max_{x,y} dist(x,y)$$

The general flavor of research in the area of Metric Distortion is to find embeddings with provably small distortion. A classical and perhaps the most important result in this area is the Johnson-Lindenstrauss (JL) Lemma [23]. The JL-Lemma says that $n$ points can be embedded in dimension $O(\epsilon^{-2}\log n)$ with distortion $1+\epsilon$. Moreover, this embedding can be found using a linear map and in randomized polynomial time. Informally, the JL Lemma says that dimension can be reduced by projecting the data along a random half-space and this projection preserves inter-point distances. This lemma has found applications in Manifold Learning, Compressed Sensing etc. besides being used for dimensionality reduction.

However, one of the limitations the JL-Lemma is the dependence of the reduced dimension on the number of points $n$. Hence as $n$ becomes very large, the number dimensions required to get a small distortion also increases. This maybe unfavorable for applications which deal with a large amount of data with large dimension (e.g. computer vision). Alon [2] showed that the dependence on $n$ is essential and the JL-Lemma is tight upto a factor of $O(\log(\frac{1}{\epsilon}))$.

Often the data is of intrinsically low dimension. In such cases, it would make more sense to reduce to a dimension which is closer to the intrinsic dimension and use methods which exploit this property of the data. Hence, recent works in the area focus on trying to remove the dependency on $n$ using the *doubling dimension* of the data [17]. Abraham et. al [1] show that a metric space $S$ of $n$ points embed into Euclidean space with dimension $O(dim(S)/\epsilon)$ and distortion $O(\log^{1+\epsilon} n)$ where $dim(S)$ denotes the doubling dimension. In this result, although the dimension is independent of $n$, however the distortion depends on $n$. One of the open problems in the area is to obtain embeddings with dimension and distortion both dependent only on $dim(S)$.

Although results are not known for the original metric $d$, but results have been obtained for the Snowflake metric $(d^\alpha)$. These metrics are obtained by raising the given metric $d$ to some fractional power$(0 < \alpha < 1)$. Gottlieb et. al [16] show that a snowflake metric can be embedded in dimension $\tilde{O}(\epsilon^{-4}dim^2(S))$ with distortion $1+\epsilon$. Bartal et. al [8] improve the above bound to $\tilde{O}(\epsilon^{-3}dim(S))$. They also provide a local dimensionality reduction in $O(\epsilon^{-2}\log k)$ dimension with distortion $1+\epsilon$ where $k$ is size of the neighborhood in which small distances are preserved.

### Our Contribution

A common thread in all the works we cited in the above section was to obtain embeddings which had a small distortion. That is, the embeddings are such that all pairs of points have small distortion. However, a less strict and more realistic requirement would be to obtain embeddings such that the distortion is small for most but not all of the points. It is fairly easy to construct examples of techniques which perform quite well in practice and have small distortion for most of the pairs of points. In Chapter 3, we construct one such example. In this work, we try to build a generic framework which can model all such dimensionality-reduction techniques.

We formalize a new probabilistic notion of what it means to preserve distance as a relaxation of Metric Distortion. We call it *Retaining Distance*. We show that common techniques like Principal Component Analysis (PCA) can be modeled by our notion. While using PCA in various applications, the following heuristic is commonly used. Choose the reduced dimension size such that PCA captures 95% or 99% of the variance. We prove that if the variance along directions orthogonal to the principal components is small then PCA retains distances. Hence, we provide theoretical justifications for something which is successfully used in practice. We give some experimental evidence which suggests that datasets on which PCA retains distances in a 'better' way, the performance on subsequent classification task is also better.

## 2.2 Probabilistic Lipschitzness

Theoretical Machine Learning works in a pessimistic or worst-case setting. For a learning algorithm to be successful, it must perform well under all possible data distributions. Often in practice, it is seen that algorithms perform much better than the bounds predicted by this analysis. One such example is seen in algorithms which try to learn from unlabeled data. Results obtained by Ben-David et al. [11], Raginsky et al. [25] prove that access to unlabeled data does not help in the worst-case. However, many applications improve their performance by taking unlabeled data into account. This is because in real life applications, the data distribution often has some nice properties which are exploited by the learning algorithm.

An assumption which is often inherent in many machine learning paradigms is that points which are 'close' to one another in the feature space tend to have same labels. To model this niceness property, a notion of Probabilistic Lipschitzness (PL) was introduced by Urner et al. [29]. Under PL Assumptions, Urner et al. [28] showed that Nearest Neighbor Algorithm has bounded sample complexity. Not only that, they show that PL assumptions lead to sample savings, i.e., faster learning from nicer distributions. They also show that under PL assumptions, proper semi-supervised learning has reduced sample complexity. Another notion which is similar to PL was introduced by Steinwart et al. [27] called the *Margin Exponent*. They use margin exponents to give learning rates for Support Vector Machines by bounding the approximation error for Gaussian Kernels.

### Our Contribution

The PL assumption essentially says that the probability that two close points have different labels is bounded and small. In this thesis, we consider a different definition of PL. We show that our new definition of PL is complementary to the previous definition. That is, there are situations in which one holds and the other does not and vice-versa. Under our PL definition, we prove that nearest neighbor has bounded sample complexity.

The crux of the thesis is in combining the two notions PL and Retaining Distance to achieve '*Information Preserving Dimensionality Reduction*'. The information that we care about in this setting is the property that close points have same labels (or PL). We show that if any distribution had PL property in the original dimension then any dimensionality reduction technique that retains distances preserves this property. Hence, the niceness property that existed in the original dimension is also present in the reduced dimension space.

# Chapter 3

# Retaining Distance

## 3.1 Preliminaries

### Framework

A domain set $S \subseteq \mathbb{R}^N$ generated i.i.d by some probability distribution $P$. Distance functions $d$ and $d'$ in $\mathbb{R}^N$ and $\mathbb{R}^n$ respectively where $N > n$. A dimensionality-reduction technique $\mathbb{DR}$ which takes points in $\mathbb{R}^N$ and returns points in $\mathbb{R}^n$. Construct a distance function $d''$ which extends $d'$ to $\mathbb{R}^N$, that is, $d''(x, y) = d'(x', y')$ where $x' = \mathbb{DR}(x)$ and $y' = \mathbb{DR}(y)$.

### Motivation

The objective of many common dimensionality-reduction techniques (or embeddings) is to 'retain' inter-point distances. Experiments also suggest that techniques which retain distances in a 'better' way perform better on subsequent classification and other tasks. Metric Distortion aims to construct embeddings which have distortion $1 + \epsilon$ $(0 < \epsilon < 1)$ for all pairs of points. Intuitively, a more realistic requirement would be ensure that the distortion is small for most but not necessarily all the pairs of points. To capture these properties, a mathematical notion of Retaining Distances is introduced.

We consider two types of events. Points which were 'close' in the original representation but whose distances have grown by a large factor (say 2) in the new representation. Another event is points whose distances have shrunk be a large factor (say 2) and as a result are 'close' in the reduced representation. Our definition essentially says that the probability of the above events is bounded and small.

**Definition 3.1** (**Retaining Distance**)**.** Consider the framework as introduced. A domain set $S$, probability distribution $P$, distance functions $d$ and $d''$. We say that a dimensionality-reduction technique $\mathbb{DR}$ retains distances when there exists functions $\psi_1$ and $\psi_2$ such that the following holds

$$\begin{aligned} \underset{x,y\sim P}{\mathbb{Pr}} \left[ d(x,y) \geq 2\, d''(x,y) \;\wedge\; d''(x,y) < \lambda \right] &\leq \psi_1(\lambda) \\ \underset{x,y\sim P}{\mathbb{Pr}} \left[ d''(x,y) \geq 2\, d(x,y) \;\wedge\; d(x,y) < \lambda \right] &\leq \psi_2(\lambda) \end{aligned}$$

**Example 3.1.** Consider points in 2-dimensional plane. Let the distribution $P$ be such that it generates points $y = 0$ and $x \in [0, 1]$ with probability $1 - \epsilon$ uniformly and points $y = 2$ and $x \in [0, 1]$ with probability $\epsilon$ uniformly. Let the dimensionality reduction technique be such that it always projects points along the $x$-axis.

Then for such a reduction technique, $\psi_1(\lambda) = \epsilon(1-\epsilon)\lambda$ and $\psi_2(\lambda) = 0$. Both these values are quite small which is consistent with our intuition that this dimensionality reduction technique performs well for the given distribution. However the distortion for this technique would be large ($> 2$) for large datasets.

## 3.2 Principal Component Analysis(PCA)

### 3.2.1 Introduction

PCA is one of the most popular dimensionality reduction techniques. PCA projects the original $N$-dimensional data along the $n$ principal directions or an $n$-dimensional linear subspace. The goal is to capture as much variance of the data as possible. The definition of PCA which we would be using is due to Hotelling [20]. Given a set of data points, project the data along orthogonal unit vectors such that the variance captured is maximum.

It turns out that finding orthonormal vectors with maximum variance captured is actually equivalent to finding the top eigenvectors of the sample covariance matrix. We omit the mathematical details of this calculation here. For a more detailed discussion interested readers are requested to refer [15] or any other standard Machine Learning text. Another way to look at PCA is that the Principal Components minimize the squared reconstruction error for the points.

From the point of view of real-world applications, the important decision here is the choice of $n$, the dimensionality of the space to which the data should be reduced. Practi-

**Algorithm 1:** Principal Component Analysis

```
Input: Data set $X = \{x_1, \ldots, x_m\}$ where each $x_i \subseteq \mathbb{R}^N$
Output: $Y = \{x'_1, \ldots, x'_m\}$ where each $x'_i \subseteq \mathbb{R}^n$, $n < N$
Compute the sample covariance matrix $S = XX^T$.
Compute the top $n$ eigenvectors of S and store in another matrix A. The matrix A
  stores the principal components of the data.
Output $Y = AX$ as the reduced representation of the original dataset $X$.
```

tioners often choose $n$ such that 'most' of the variance of the data is captured. A dimension $n$ which retains 99% of the variance is often considered a good choice.

Intuitively, capturing the variance seems equivalent to capturing the relevant variations and information inherent in the data. We will prove in the next subsection that this in fact relates to distances. We will prove that as long as 'most' of the variance of the data is captured by the $n$ principal components then PCA *retains distances*. We will quantify 'most' of the variance as being such that the variance along any other direction orthogonal to the principal component is bounded by $\epsilon$ for some small constant $\epsilon$.

### 3.2.2 PCA retains distance

Let us assume that PCA projects the data along the $n$ orthogonal unit vectors given by $v_1, \ldots, v_n$. Let $v_{n+1}, \ldots, v_N$ be some other vectors such that $v_1, \ldots, v_N$ form an orthogonal basis for the original $N$-dimensional space. Then PCA is equivalent to choosing the first $n$ dimensions amongst the $N$ given dimensions in this space. Let the distance functions be the standard Euclidean distances. Hence, in this framework $d(x, y) = \|x - y\| = \sqrt{(x_1 - y_1)^2 + \ldots + (x_N - y_N)^2}$ and $d''(x, y) = \sqrt{(x_1 - y_1)^2 + \ldots + (x_n - y_n)^2}$.

Before we prove that PCA retains distances, let us consider the following lemma which will be useful later on. In order to prove the lemma below, we will use a classical result from Probability Theory called the Chebyshev's inequality. Chebyshev's inequality essentially says that a random variable cannot take values which are very 'far' from the mean. More formally, the probability that a random variable takes values far from the mean is bounded.

**Chebyshev's inequality** [24] Let $x$ be any random variable with expected value $\mu$ and variance $\sigma^2$. Let $a > 0$ be any real number,

$$\Pr_{x \sim P} \left[ |x - \mu| \geq a\,\sigma \right] \leq \frac{1}{a^2} \tag{3.1}$$

**Lemma 3.1.** *Let $S \subseteq \mathbb{R}^N$ be an unlabeled data set generated i.i.d by some unknown probability distribution $P$. Let $P_i$ be the projection of $P$ in direction $i, 1 \leq i \leq N$. If $var(P_i) < \epsilon, \forall\, n+1 \leq i \leq N$, then $\Pr_{x,y \sim P} \left[ d(x,y) - d''(x,y) \geq \lambda \right] \leq \frac{8\,(N-n)^2\,\epsilon^2}{\lambda^2}$.*

*Proof.* In our proof, we will frequently use the following. If $A \subseteq B$ then $\Pr(A) \leq \Pr(B)$. Now, observe that $d(x,y) - d''(x,y) \geq \lambda \implies d^2(x,y) \geq (d''(x,y) + \lambda)^2 \implies d^2(x,y) - d''^2(x,y) \geq \lambda^2$.

$$\therefore \Pr_{x,y \sim P} \left[ d(x,y) - d''(x,y) \geq \lambda \right] = \Pr_{x,y \sim P} \left[ \sum_{i=n+1}^{N} (x_i - y_i)^2 \geq \lambda^2 \right]$$

Now observe that $\displaystyle\sum_{i=n+1}^{N} (x_i - y_i)^2 \geq \lambda^2 \implies \exists i : (x_i - y_i)^2 \geq \frac{\lambda^2}{N-n}$

$$\therefore \Pr_{x,y \sim P} \left[ \sum_{i=n+1}^{N} (x_i - y_i)^2 \geq \lambda^2 \right] \leq \Pr_{x,y \sim P} \left[ \exists i : (x_i - y_i)^2 \geq \frac{\lambda^2}{N-n} \right]$$

$$\leq \sum_{i=n+1}^{N} \Pr_{x,y \sim P} \left[ |x_i - y_i| \geq \frac{\lambda}{\sqrt{N-n}} \right]$$

$$\leq \sum_{i=n+1}^{N} \Pr_{x,y \sim P} \left[ |x_i - \mu_i| + |y_i - \mu_i| \geq \frac{\lambda}{\sqrt{N-n}} \right], \text{ where } \mu_i = \text{mean}(P_i)$$

As before, one of $|x_i - \mu_i|$ or $|y_i - \mu_i| > \dfrac{\lambda}{2\sqrt{N-n}}$ and we get

$$\leq \sum_{i=n+1}^{N} \Pr_{x,y \sim P} \left[ |x_i - \mu_i| \geq \frac{\lambda}{2\sqrt{N-n}} \right] + \Pr_{x,y \sim P} \left[ |y_i - \mu_i| \geq \frac{\lambda}{2\sqrt{N-n}} \right]$$

$= 2 \displaystyle\sum_{i=n+1}^{N} \Pr_{x_i \sim P_i} \left[ |x_i - \mu_i| \geq \frac{\lambda}{2\sqrt{N-n}} \right]$. Using, Chebyshev's Inequality (Eqn 3.1), we get

$$\leq \sum_{i=n+1}^{N} \frac{8\,(N-n)\,var^2(P_i)}{\lambda^2} \leq \frac{8\,(N-n)^2\,\epsilon^2}{\lambda^2}$$

□

Denote $f_1(\lambda) := \underset{x,y\sim P}{\mathbb{Pr}}\,[d(x,y) \geq 2\,d''(x,y) \;\wedge\; d''(x,y) < \lambda]$ and $f_2(\lambda) := \underset{x,y\sim P}{\mathbb{Pr}}\,[d''(x,y) \geq 2\,d(x,y) \;\wedge\; d(x,y) < \lambda]$. To prove that PCA retains distances, we need to show that $f_1(\lambda) \leq \psi_1(\lambda)$ and $f_2(\lambda) \leq \psi_2(\lambda)$ for some known functions $\psi_1$ and $\psi_2$. We will now use Lemma 3.1 to prove the following theorem which establishes that PCA retains distance.

**Theorem 3.2.** *Let the framework be as in Lemma 3.1. Unlabeled dataset $S \subseteq \mathbb{R}^N$ generated i.i.d by some unknown probability distribution $P$ and $var(P_i) < \epsilon, \forall\, n+1 \leq i \leq N$. Let $f_1$ and $f_2$ be functions as defined above. In addition, let $f_1(t) = c$ for some constant $t$. Then PCA retains distances with $\psi_2(\lambda) = 0$ and $\psi_1(\lambda) = c + \frac{512\,(N-n)^2\,\epsilon^2}{3t^2}$. In other words, $\psi_1(\lambda) \in O(\epsilon^2)$*

*Proof.* In case of PCA, note that $d'' < d$. Hence, for all $\lambda$, $f_2(\lambda) = 0$ and hence $\psi_2(\lambda) = 0$. We will now try to obtain an upper bound $f_1$.

Now, observe that $f_1$ is an increasing function of $\lambda$. Therefore, for $\lambda \leq t$, we have that $f_1(\lambda) \leq f_1(t) = c < \psi_1(\lambda)$. For $\lambda > t$, we get,

$$f_1(\lambda) = f_1\left(\frac{\lambda}{2}\right) + \underset{x,y\sim P}{\mathbb{Pr}}\left[d(x,y) \geq 2\,d''(x,y) \;\wedge\; \left(\frac{\lambda}{2} < d''(x,y) < \lambda\right)\right]$$

$$\implies f_1(\lambda) \leq f_1\left(\frac{\lambda}{2}\right) + \underset{x,y\sim P}{\mathbb{Pr}}\left[d(x,y) - d''(x,y) \geq \frac{\lambda}{2}\right]$$

Let $c_1 := 32(N-n)^2\epsilon^2$, and using Lemma 3.1, we get that

$$f_1(\lambda) \leq f_1\left(\frac{\lambda}{2}\right) + \frac{c_1}{\lambda^2}$$
$$\leq f_1\left(\frac{\lambda}{4}\right) + \frac{c_1}{\lambda^2}[1^2 + 2^2] \leq f\left(\frac{\lambda}{8}\right) + \frac{c_1}{\lambda^2}[1^2 + 2^2 + 2^4]$$

Now let $m$ be such that $\frac{\lambda}{2^{m+1}} < t \leq \frac{\lambda}{2^m}$. Then, repeating the above steps we get that

$$f_1(\lambda) \;\leq\; f_1\left(\frac{\lambda}{2^{m+1}}\right) + \frac{c_1}{\lambda^2}[1^2 + 2^2 + 2^4 + \ldots + 2^{2m}]$$
$$\leq\; c + \frac{c_1}{\lambda^2}\sum_{i=0}^{m} 4^i \;\leq\; c + \frac{c_1}{\lambda^2}.\frac{4^{m+2}-4}{3} \;\leq\; c + \frac{16\,c_1\,4^m}{3\lambda^2}$$
$$=\; c + \frac{512(N-n)^2\epsilon^2}{3t^2} =: \psi_1(\lambda)$$

□

### Summary

We see that $\psi_1(\lambda) = O(\epsilon^2)$ and $\psi_2(\lambda) = 0$. Hence, if the eigenvectors capture 'most' of the variance of the data such that the variance along directions orthogonal to the eigenvectors is small then PCA retains the distances between two points. Thus, our notion of retaining distance is able to model PCA using assumptions which are actually used in practice. Note that, the metric distortion of PCA would be large. Since for some pairs of points the distortion maybe large. However, for most of the pairs of points the distortion is small. This provides strong evidence that our notion is more realistic and closer to what is seen in real-world applications.

# Chapter 4

# Probabilistic Lipschitzness(PL)

## 4.1 Preliminaries

### Motivation

The framework of Statistical Learning Theory is agnostic to the probability distribution which generates the data. An algorithm is said to be learnable only when it has low error over all possible distributions which generate the data. Thus, learning theory works in the pessimistic or the worst-case scenario. However, in practical scenarios, the data often does not exhibit worst-case behavior.

The success of many algorithms can be attributed to the fact that the distribution has certain nice properties. For example, if a distribution is such that the closest point to any given point has opposite label then for such a distribution Nearest Neighbor would have large error and would probably not be learnable. Similarly, if there exists no half-space which separates the data points with low error, a linear classifier would have large error and would not be learnable.

The notion of Probabilistic Lipschitzness was introduced by Urner et. al in [29] to quantify how likely it is for two close points to have same labels. "PL is useful for modeling niceness properties of distributions with deterministic labeling functions [28]." In a nutshell, the PL assumption says that the probability of two close points having different labels is bounded and small. Such a relation is inherent in many Machine Learning paradigms.

In this thesis, we consider a slightly different definition of PL. We will show that the sample complexity bounds for Nearest Neighbor that were obtained using the original definition can also be obtained using our definition. Our definition of PL is more useful in the context of a distance retaining dimensionality reduction (This we will show in Chapter 5). Throughout this Chapter and the remainder of the thesis, we will refer to our definition of PL as *PL-Conditional* and the original definition as *PL-Unary*.

### Framework

Let $S$ be an unlabeled set generated i.i.d by some probability distribution $P$ and labeled by some deterministic function $l$. Let $d$ be a distance function.

**Definition 4.1** (**PL-Unary**)**.** The labeling function $l$ satisfies Probabilistic Lipschitzness w.r.t to the Urner definition (call it PL-Unary) when there exists a function $\phi$ such that

$$\Pr_{x\sim P}\left[\exists y : d(x,y) < \lambda \ \wedge \ l(x) \neq l(y)\right] \ \leq \ \phi(\lambda)$$

In the papers where PL-Unary was introduced, two alternate formulations of the PL-Unary definition have been considered. The work in [29] considers the above definition. A more recent version of the author's work on PL [28] considers another definition which is almost similar (very slightly stronger) to the original definition.

$$\Pr_{x\sim P}\left[\Pr_{y\sim P}\left[d(x,y) < \lambda \ \wedge \ l(x) \neq l(y)\right] > 0\right] \ \leq \ \phi(\lambda) \tag{4.1}$$

The sample complexity bounds and other results they present hold for both the versions of this definition. This slight distinction between the two versions will become more apparent when we consider the following example.

**Example 4.1.** Let the domain be $\mathbb{X} = [0,1]$, the labeling function $l$ be such that it labels 1 for irrationals and 0 for rationals and $P$ be the uniform distribution. Then given any point $x$ and any $\lambda$, there always exists another point $y$ such that $d(x,y) < \lambda$ and $l(x) \neq l(y)$. Hence, according to the first version of PL-Unary, $\phi(\lambda) = 1$. However, note that on the real line, the probability of generating a rational number is 0. Hence, according to the second version of PL-Unary, $\phi(\lambda) = 0$.

In this chapter and the remainder of the thesis, we will be using the second version (Equation 4.1) as the definition of PL-Unary.

**Definition 4.2** (**PL-Conditional**)**.** We say that the labeling function $l$ satisfies PL-Conditional when there exists a function $\phi$ such that

$$\Pr_{x,y\sim P}\left[l(x) \neq l(y) \mid d(x,y) < \lambda\right] \ \leq \ \phi(\lambda)$$

## 4.2 Sample Complexity bounds for Nearest Neighbor under PL-Conditional

We will now prove sample complexity bounds for Nearest Neighbor (NN) under PL-Conditional assumptions. Nearest Neighbor is perhaps the most simplest to state algorithms in Machine Learning. Let $S$ be any training set of size $m$. Now, given any query point $x$, find the point in $S$ which is closest to $x$. Call this point the Nearest Neighbor of $x$ in S denoted by $\pi_S(x)$. Label the point $x$ with the same label as that of $\pi_S(x)$.

**Theorem 4.1.** *Let $S \subseteq [0,1]^N$ be an unlabeled set generated i.i.d by some distribution $P$. Let $S$ be labeled by some deterministic function $l$ which satisfies PL-Conditional with function $\phi$ . Then the sample complexity of Nearest Neighbor $m_{NN}$ is upper bounded by*

$$m_{NN}(\epsilon, \delta) \quad \leq \quad \frac{2}{\epsilon\delta e} \cdot \left( \frac{\sqrt{N}}{\phi^{-1}(\epsilon\delta/2)} \right)^N \tag{4.2}$$

*Outline of the proof*: The full proof of the theorem is quite long (about two and a half pages in length). We give only the proof ideas in this section and include the detailed proof in the Appendix A.

The essential proof idea is the same as that used in the proof of Theorem 19.3 in [26]. We divide the region $[0,1]^N$ into axis-aligned hyper rectangular boxes each of diagonal length $\lambda$. Now given a training set $S$ of size $m$, there are two possibilities. A query point $x$ lies in a box that contains a point from $S$ or the point lies in a box that is empty.

In the first case, since the box already contains a point from $S$, we know that the distance to the Nearest Neighbor of $x$ is bounded by the box length $\lambda$. We then use PL-Conditional to bound the probability of error in this case as $\phi(\lambda)$. The second case is equivalent to generating $m$ points such that none of the points lie in a given box. We use Lemma A.1 to bound the probability of this event. The basic intuition is that if the training set size $m$ is large enough, then most of the boxes would be hit by a sample point and the probability of a box being empty would be small. Thus, we see that Nearest Neighbor has bounded error in this case. The more examples we get the lesser is the probability of error.

## 4.3 Extensions to $k$-Nearest Neighbor

In this section we extend our results to a more generalized version of Nearest Neighbor, that is, $k$-Nearest Neighbor (kNN). Given any training set $S$ and a query point $x$, instead of just the closest point, find $k$ points in $S$ which are closest to $x$. Then assign the label of $x$ as the majority label over these $k$ points. We will now prove sample complexity bounds for kNN under both PL-Conditional and PL-Unary assumptions.

**Theorem 4.2.** *Let $S \subseteq [0,1]^N$ be an unlabeled set generated i.i.d by some distribution $P$. Let $S$ be labeled by some deterministic function $l$ which satisfies PL-Conditional with function $\phi$ . Then the sample complexity of $k$-Nearest Neighbor $m_{kNN}$ is upper bounded by*

$$m_{kNN}(\epsilon, \delta) \quad \leq \quad \frac{4k}{\epsilon\delta} \cdot \left( \frac{\sqrt{N}}{\phi^{-1}(\epsilon\delta/4)} \right)^N \tag{4.3}$$

*Proof.* The essential ideas used in the proof are identical to that used in Theorem 4.1. Divide the region into axis-aligned boxes of predefined sizes. The basic idea is that if we have enough samples we can hit most of the boxes $k$ times. Now, either the nearest neighbors are such that all of them are at a distance less than $\lambda$ or there exists atleast one with distance greater than $\lambda$. Using PL, we bound the probability of error in the first case. The second case has a small probability of occurrence (Lemma A.2) The detailed proof is given in Appendix A. □

We will now prove sample complexity bounds for kNN when the labeling function satisfies the PL-Unary assumption. As expected, the proof is a simple extension of the proof for Nearest Neighbor given by Urner et. al [28].

**Theorem 4.3.** *Let $S \subseteq [0,1]^N$ be an unlabeled set generated i.i.d by some distribution $P$. Let $S$ be labeled by some deterministic function $l$ which satisfies PL-Unary with function $\phi$ . Then the sample complexity of $k$-Nearest Neighbor $m_{kNN}$ is upper bounded by*

$$m_{kNN}(\epsilon, \delta) \quad \leq \quad \frac{4k}{\epsilon\delta} \cdot \left( \frac{\sqrt{N}}{\phi^{-1}((\epsilon\delta/2)^{2/k})} \right)^N \tag{4.4}$$

*Proof.* Please refer to the appendix. □

## 4.4 PL-Unary vs PL-Conditional

We will now compare the two notions of PL and see how they compare against one another. We want to compare them both theoretically and practically. In this section, we only consider the theoretical aspects. Detailed experiments are given Chapter 7.

The first question that we try to investigate is that whether one of notions is stronger than the other. We will construct two examples which will show that none is always stronger than the other. Thus, there are scenarios when PL-Conditional is better and scenarios when PL-Unary is better.

Note that both the definitions of Probabilistic Lipschitzness are parametrized by a function $\phi$. Denote by $\phi_{PLC}$ the function which parametrizes our definition and by $\phi_{PLU}$ the function which parametrizes the other definition.

**Example 4.2.** (**PL-Conditional but not PL-Unary**)

Consider the real line from $[0, 1]$. Let $\lambda$ be some constant and $\gamma$ be another constant such that $\gamma << \lambda$. Consider the closed interval $X_k = [k\lambda - \gamma, k\lambda + \gamma]$ for some $k \in \mathbb{N}$. Let the domain be the union of all such closed intervals which lie completely within $[0, 1]$. That is the domain $\mathbb{X} = \cup X_k$ for all $k$ such that $X_k \subseteq [0, 1]$. Hence, the domain is made up of $n$ subintervals where $\frac{1}{\lambda} - 1 \leq n \leq \frac{1}{\lambda}$.

Let the labeling function be such that all points of the form $k\lambda$ where $k \in \mathbb{N}$ are labeled 1 and all the other points are labeled 0. Now, consider the following distribution $P$ over the domain $\mathbb{X}$. All the subintervals are given equal weight of $\frac{1}{n}$. Within the subinterval, the point of the form $k\lambda$ (center) is given a weight of $\frac{1-\gamma}{n}$ and the remaining weight of $\frac{\gamma}{n}$ is spread uniformly over the remaining points.

Now, observe that, given any point $x$ there exists another point $y$ with finite probability such that $d(x, y) < \lambda$ and $l(x) \neq l(y)$. Hence, $\phi_{PLU} = 1$. We now need to show that for the above example $\phi_{PLC}$ is small.

**Lemma 4.4.** *For the example described above, $\phi_{PLC}(\lambda) = \frac{4\gamma}{\lambda}$.*

*Proof.* For the proof of the above lemma, we need a small result from probability theory. $Pr(A \cap B|C) \leq \frac{Pr(B \cap C|A)}{Pr(C|A)}$. The proof of this result is fairly elementary. Now, $l(x) \neq l(y)$ if and only if $x$ is some interval center ($C_i$) and $y$ is not a center (denoted by $NC$) or vice-versa. Hence, we get

$$\underset{x,y \sim P}{\mathbb{Pr}} \left[ l(x) \neq l(y) \mid d(x, y) < \lambda \right] = 2 \sum_i \underset{x,y \sim P}{\mathbb{Pr}} \left[ x = C_i \ \wedge \ y \text{ is } NC \mid d(x, y) < \lambda \right]$$

Now, consider the quantity on the right of the above equation and using the probability result, we get that

$$\underset{x,y \sim P}{\mathbb{Pr}} \left[ x = C_i \ \wedge \ y \text{ is } NC \mid d(x, y) < \lambda \right] = \frac{\underset{x,y \sim P}{\mathbb{Pr}} \left[ d(x, y) < \lambda \ \wedge \ y \text{ is } NC \mid x = C_i \right]}{\underset{x,y \sim P}{\mathbb{Pr}} \left[ d(x, y) < \lambda \mid x = C_i \right]}$$

In the above equation, denote the numerator by $num(C_i)$ and the denominator by $den(C_i)$. Now, there are two possibilities. The first possibility is when the interval is not one of the first or the last intervals. And the second possibility is when it is one of the two.

In the former case, we have that $num(C_i) = \frac{4\gamma}{2n}$ and $den(C_i) = \frac{4\gamma}{2n} + \frac{2(1-\gamma)}{2n}$. And hence, we get that $\frac{num(C_i)}{den(C_i)} < \frac{4\gamma}{4+2\gamma} < 2\gamma$. Similarly, in the latter case, we get that $num(C_i) = \frac{3\gamma}{2n}$ and $den(C_i) = \frac{3\gamma}{2n} + \frac{2(1-\gamma)}{2n}$. And hence, we get that $\frac{num(C_i)}{den(C_i)} < \frac{3\gamma}{2+\gamma} < 2\gamma$. Substituting this throughout, we get that

$$\underset{x,y\sim P}{\mathbb{P}\mathrm{r}} \left[ l(x) \neq l(y) \mid d(x,y) < \lambda \right] \;<\; 2\sum_i 2\gamma \;=\; 4\gamma n \;<\; \frac{4\gamma}{\lambda}$$

□

Since $\gamma << \lambda$, we see that $\phi_{PLC}$ is small as desired. However, in this case $\phi_{PLU} = 1$. Hence, we got an example as desired. Now, we will construct an example which is the other way round. That is, PL-Unary is small but PL-Conditional is large.

**Example 4.3.** (**PL-Unary but not PL-Conditional**)

Consider the real line from $[0,1]$. Let $\lambda << 1$ be some small constant. Let $S = \{\lambda, \ldots, k\lambda\}$ be the maximal set such that $k \in \mathbb{N}$ and $S \subseteq [0,1]$. Let the domain $\mathbb{X} = S \cup (k + \frac{1}{2}\lambda)$. Let $n$ be the number of points in the domain $\mathbb{X}$. Then $\frac{1}{\lambda} \leq n \leq \frac{1}{\lambda} + 1$. Let $P$ be the uniform distribution over the domain $\mathbb{X}$. Let the labeling function be such that it labels all points in $S$ as 1 and the point $(k + \frac{1}{2}\lambda)$ as 0.

Now, if it is given that $d(x,y) < \lambda$, then the only possibility is $x = k\lambda$ and $y = (k+\frac{1}{2})\lambda$ or vice-versa. In both of these cases due to the choice of the labeling function, we know that $l(x) \neq l(y)$. Hence, $\underset{x,y\sim P}{\mathbb{P}\mathrm{r}} \left[ l(x) \neq l(y) \mid d(x,y) < \lambda \right] = 1 = \phi_{PLC}(\lambda)$. In this example, $\phi_{PLC}$ has a large value. The only thing left to show is that $\phi_{PLU}$ has a small value in this case. For points $x \in S \setminus k\lambda$, we see that the $\underset{y\sim P}{\mathbb{P}\mathrm{r}} \left[ d(x,y) < \lambda \right] = 0$. For the other two points, we see that there does exist a $y$ with non-zero probability of opposite label. Hence, we get that $\underset{x\sim P}{\mathbb{P}\mathrm{r}} \left[ \underset{y\sim P}{\mathbb{P}\mathrm{r}} \left[ d(x,y) < \lambda \,\wedge\, l(x) \neq l(y) \right] > 0 \right] \;=\; \frac{2}{n} \;\leq\; 2\lambda = \phi_{PLU}(\lambda)$. For this example, we see that $\phi_{PLU}$ has a small value but $\phi_{PLC}$ does not.

In both the Examples (4.2 and 4.3), intuitively we had the situation that close points had same labels. In Example 4.3, PL-Unary was better at modeling this intuition since it had a small value while in Example 4.2 PL-Conditional was better. Hence, we conclude that none of the two notions of Probabilistic Lipschitzness is always than the other. We would also like to see how these two notions compare in practice. From our experiments on different datasets, it seems that the performance of both the notions are somewhat similar. The details of our experiments are included in Chapter 7.

The last thing that we want to consider is the convergence rates for Nearest Neighbor Learning ($m_{NN}$) under PL-Unary and PL-Conditional. From Theorem 4.1, we know that under PL-Conditional

$$m_{NN} \leq \frac{2}{\epsilon\delta e}.\left(\frac{\sqrt{N}}{\phi^{-1}_{PLC}(\epsilon\delta/2)}\right)^{N}.$$

Similarly, Theorem 3 of [28] says that under PL-Unary, the sample complexity upper bound for Nearest Neighbor

$$m_{NN} \leq \frac{2}{\epsilon\delta e}.\left(\frac{\sqrt{N}}{\phi^{-1}_{PLU}(\epsilon/2)}\right)^{N}.$$

Hence, given $\epsilon$ and $\delta$ the learning rates depend on the values of $\phi^{-1}_{PLC}(\epsilon\delta/2)$ and $\phi^{-1}_{PLU}(\epsilon/2)$. Since, nothing can be said about whether $\phi_{PLC}$ is greater or less than $\phi_{PLU}$ in general, we cannot say anything about the learning rates in general. The actual rates would vary from application to application. In some cases, learning under PL-Conditional would be faster while in some learning under PL-Unary.

# Chapter 5

# Distance Retaining Reduction preserves PL-Conditional

So far, we have introduced the notion of Retaining Distance and demonstrated in Chapter 3 that it is good at modeling dimensionality reduction techniques which preserve inter-point distances. In this chapter, we will show that this is not the only advantage of a *distance retaining* reduction.

Besides inter-point distances, a distance retaining dimensionality reduction preserves some of the *nice* properties of the distribution as well. One of such nice properties is Probabilistic Lipschitzness which we introduced in Chapter 4. Theorem 5.1 and Lemma 5.2 together show that if we have PL-Conditional in the original dimension and any dimensionality reduction technique which retains distance reduces the data to a lower dimension then we would have the PL-Conditional property in the lower dimension as well.

Another way of stating the same result is that, Nearest Neighbor has bounded sample complexity in the reduced dimension space as well. There are certain conditions under which Nearest Neighbor has bounded sample complexity. We want to show that under a distance retaining reduction, if those conditions are true in the original dimension, then those conditions are true in the reduced dimension as well. One such condition is PL-Conditional.

PL-Conditional definition says that $\underset{x,y\sim P}{\mathbb{Pr}}\left[l(x) \neq l(y) \mid d(x,y) < \lambda\right] \leq \phi(\lambda)$. We break this definition into two statements. The first is that the probability of the 'and' condition is upper bounded and that the denominator is lower bounded by some function of $\lambda$. More formally, we assume that $\underset{x,y\sim P}{\mathbb{Pr}}\left[l(x) \neq l(y) \wedge d(x,y) < \lambda\right] \leq \alpha(\lambda)$ and

$\underset{x,y\sim P}{\mathbb{Pr}}\left[d(x,y) < \lambda\right] \;\geq\; \beta(\lambda)$ where $\phi = \frac{\alpha}{\beta}$. We will now show that these quantities are bounded in the reduced dimension space as well.

**Theorem 5.1.** *Consider the framework as in the Definition 3.1. Domain set $X \subseteq \mathbb{R}^N$ generated by some probability distribution $P$, distance functions $d$ and $d''$, dimensionality reduction technique $\mathbb{DR}$ which retains distances wrt functions $\psi_1$ and $\psi_2$ and a labeling function $l$. Denote $x' = \mathbb{DR}(x)$ and $y' = \mathbb{DR}(y)$. Let $P'$ be the distribution obtained by projecting $P$, that is, $P'(x') = P(x)$ and $l'$ be the corresponding labeling function, that is, $l'(x') = l(x)$. If $\underset{x,y\sim P}{\mathbb{Pr}}\left[l(x) \neq l(y) \wedge d(x,y) < \lambda\right] \;\leq\; \alpha(\lambda)$ then $l'$ satisfies the corresponding inequality in low dimension $\underset{x',y'\sim P}{\mathbb{Pr}}\left[l'(x') \neq l'(y') \wedge d(x',y') < \lambda\right] \;\leq\; \alpha_1(\lambda)$ where $\alpha_1 = \psi_1 + \alpha.2$.*

*Proof.* We know that $\mathbb{DR}$ retains distances, hence

$$\underset{x,y\sim P}{\mathbb{Pr}}\left[d(x,y) \geq 2d''(x,y) \wedge d''(x,y) < \lambda\right] \;\leq\; \psi_1(\lambda)$$

Note, that $l', d'$ and $P'$ are just projections of $l, d''$ and $P$ respectively. Hence, we have

$$\begin{aligned}
&\underset{x',y'\sim P'}{\mathbb{Pr}}\left[d'(x',y') < \lambda \;\wedge\; l'(x) \neq l'(y)\right] \;=\; \underset{x,y\sim P}{\mathbb{Pr}}\left[d''(x,y) < \lambda \;\wedge\; l(x) \neq l(y)\right] \\
&= \underset{x,y\sim P}{\mathbb{Pr}}\left[d(x,y) \leq 2d''(x,y) \wedge d''(x,y) < \lambda \wedge l(x) \neq l(y)\right] \\
&\qquad + \underset{x,y\sim P}{\mathbb{Pr}}\left[d(x,y) > 2d''(x,y) \;\wedge\; d''(x,y) < \lambda \;\wedge\; l(x) \neq l(y)\right] \\
&\leq \underset{x,y\sim P}{\mathbb{Pr}}\left[d(x,y) < 2\lambda \;\wedge\; l(x) \neq l(y)\right] \;+\; \underset{x,y\sim P}{\mathbb{Pr}}\left[d(x,y) > 2d''(x,y) \;\wedge\; d''(x,y) < \lambda\right] \\
&\leq \alpha(2\lambda) \;+\; \psi_1(\lambda) =: \; \alpha_1(\lambda)
\end{aligned}$$

□

To complete the proof that the labeling function satisfies PL-Conditional in the new dimension as well, we need to lower bound $\mathbb{Pr}[d''(x,y) < \lambda]$. The proof for this is fairly easy as is stated as a Lemma below.

**Lemma 5.2.** *Consider the framework as in Theorem 5.1. If $\underset{x,y\sim P}{\mathbb{Pr}}\left[d(x,y) < \lambda\right] \geq \beta(\lambda)$, then $\underset{x,y\sim P}{\mathbb{Pr}}\left[d''(x,y) < \lambda\right] \geq \beta_1(\lambda)$ where $\beta_1 = \beta.\frac{1}{2} - \psi_2$.*

*Proof.*

$$\underset{x,y\sim P}{\mathbb{Pr}}\left[\, d(x,y) < \frac{\lambda}{2}\,\right] - \psi_2(\lambda) < \underset{x,y\sim P}{\mathbb{Pr}}\left[\, d''(x,y) < \lambda\,\right] < \underset{x,y\sim P}{\mathbb{Pr}}\left[\, d(x,y) < 2\lambda\,\right] + \psi_1(\lambda)$$

□

We divided the PL-Conditional definition as a bound on two quantities. Theorem 5.1 and Lemma 5.2 together show that if these quantities are bounded in the higher dimension then they are bounded in the lower dimension as well provided that the reduction retains distance. In other words, PL-Conditional holds in the reduced dimension as well. Thus Nearest Neighbor bounds and other nice properties follow in the reduced dimension space.

In our discussion in this chapter, one slight detail is missing. Note that Probabilistic Lipschitzness is sensitive to the scale of the data. To obtain bounds for Nearest Neighbor under PL, we needed an implicit assumption on the diameter of the data. This is because we assumed that the domain was $[0,1]^N$. Hence, we should upper bound the diameter in the reduced dimension as well. However, in this work we only considered dimensionality reduction techniques which reduce the distance namely PCA (where diameter reduces), hence we don't give any explicit upper bounds. In future, we plan to extend our approach to other reduction techniques. In such a case, these bounds would be essential.

# Chapter 6

# Applications

So far we have presented a theoretical analysis of the different niceness notions. Specifically, we showed that our notion of Retaining Distance is good at modeling dimensionality reduction techniques which preserve inter-point distances. Moreover, such a reduction also preserves Probabilistic Lipschitzness (another nice property of a distribution). In this section, we want to show how our notion of Retaining Distance might be used in practice. Consider the algorithm below which can be used to do an empirical evaluation of a reduction technique.

**Algorithm 2:** Empirical evaluation of a reduced representation

```
Input: Unlabelled data set $X \subseteq \mathbb{R}^N$ and $X' \subseteq \mathbb{R}^n$.
Output: $f_1(\lambda)$ and $f_2(\lambda)$ for different values of $\lambda$.
Randomly sample $n$ (say $100,000$) pairs of points. For each of the pairs of points,
  compute the original distance $d(x, y)$ and the reduced distance $d''(x, y)$.
Let $f_1(\lambda)$ = Fraction of pairs of points $[\, d(x, y) \geq 2\, d''(x, y) \wedge d''(x, y) < \lambda \,]$ .
Let $f_2(\lambda)$ = Fraction of pairs of points $[\, d''(x, y) \geq 2\, d(x, y) \wedge d(x, y) < \lambda \,]$.
Compute $f_1$ for different values of $d''(x, y)$ and $f_2$ for different values of $d(x, y)$.
```

From our definition of Retaining Distance (Defn. 3.1), a good dimensionality reduction technique should have low values for functions $\psi_1$ and $\psi_2$. However, calculation of $\psi_1$ and $\psi_2$ needs knowledge of the distribution which is almost always not available to us in practical situations. Hence, we replace the 'probability over $x, y$' used in definition 3.1 by 'fraction of pairs of points' and introduce quantities $f_1$ and $f_2$.

For representations that preserve distance, we would expect $f_1$ and $f_2$ values to be small and bounded especially for small values of $\lambda$. Our experiments (Section 7.1) indicate that there is some correlation between the performance of a reduced representation and values of $f_1$ and $f_2$. The lower these values the better is the performance (classification accuracy) of the reduced representation. Thus, our notion is somewhat helpful in capturing the usefulness of a representation especially in the context of PCA.

# Chapter 7

# Experiments

We now describe the experiments we ran to validate and compare the different notions we introduced in this thesis. Our experiments can broadly be divided into two categories. The first set of experiments is used to validate the notion of Retaining Distance. In the second set of experiments we compare how the two notions of Probabilistic Lipschitzness, PL-Unary and PL-Conditional, perform in practice. We ran all our experiments on a standard Linux distribution running Ubuntu 13.04 with 4GBs Main Memory. We have used Octave (an open source version of Matlab) to implement all our algorithms.

Before we discuss our experimental results in detail, we first describe each of the datasets on which we ran our experiments.

- *MNIST* is a dataset of images of handwritten digits. Each image is $28 * 28$, hence the data is 784-dimensional. The dataset has $60,000$ training examples and a test set of $10,000$ examples.

- *Gassensor* is our abbreviation for the Gas Sensor Array Drift Dataset [31]. This dataset contains measurements from chemical sensors used to differentiate between six gases at different levels of concentrations. The dataset is 128-dimensional and has 13910 examples. We randomly divided these examples into train and test sets of sizes $12,000$ and $1,910$ respectively.

- *HAR* is our abbreviation for Human Activity Recognition using Smartphones dataset [3]. This dataset contains recordings of subjects while performing six different activities. The dataset is 561-dimensional and contains 7352 train and 2947 test examples.

- *Spambase* is a dataset of emails classified as spam or not spam available on the UCI Machine Learning Repository [7]. It contains 4601 examples each of which have a dimension of 57. We randomly split the examples into a training set of size 3900 and a test set of size 701.
- *Isolet* is a dataset of audio recordings of the 26 english letters available on UCI Repository [7]. It contains about 7800 examples of recordings each of which have dimension 617. We randomly split the dataset into a training and a test set of size 6238 and 1559 respectively.

A point to note here is that we have tried to include datasets from diverse domains. One of our datasets is from the domain of speech recognition, one from image recognition, one from chemical sensors, one from email classification and another one from activity recognition. We used such a collection of datasets so that our results and conclusions would be more general in nature and not restricted to a particular domain.

## 7.1 Retaining Distance

In Chapter 3, we introduced our notion of Retaining Distance. We showed that this notion is good at modeling dimensionality reduction techniques which preserve inter-point distances. Specifically, we proved that PCA retains distance if the principal components capture most of the variance of the distribution. We want to validate this experimentally as well.

This is how our experiments work. We take a dataset and use PCA to reduce the dimension of the data in such a way that about 98% of the sample variance is preserved. This is the reduced representation of a dataset. Our goal is to show that the Retaining Distance correlates to the performance of the reduced representations. To show that PCA retains distance, we need to show that the probabilities introduced in Definition 3.1 are small. Since, we have no knowledge of the distribution which generates the data, we use the quantities $f_1$ and $f_2$ (introduced in Chapter 6) to estimate these probabilities.

$$\begin{aligned} f_1(\lambda) &= \text{Fraction of pairs of points}\,[\,d(x,y) \geq 2\,d''(x,y) \wedge d''(x,y) < \lambda\,] \\ f_2(\lambda) &= \text{Fraction of pairs of points}\,[\,d''(x,y) \geq 2\,d(x,y) \wedge d(x,y) < \lambda\,] \end{aligned}$$

We want to compare the $f_1$ and $f_2$ as a function of $\lambda$ for different datasets. Hence it makes sense to have the distance functions $d$ and $d''$ on the same scale for all the datasets.

Table 7.1: Difference of classification accuracy (test) on reduced (PCA) representation vs Original representation on different datasets over 4 different algorithms

**MNIST**

| | Algorithm | | | | Average Accuracy Gap |
|---|---|---|---|---|---|
| | Nearest Neighbor | Regression | Linear SVM | Gaussian SVM | |
| PCA | 96.92 | 92.19 | 94.64 | 98.62 | **0.00** |
| Original | 96.91 | 92.2 | 94.69 | 98.59 | |
| **Gassensor** | | | | | |
| PCA | 99.319 | 94.398 | 87.43 | 99.27 | **2.81** |
| Original | 99.267 | 95.602 | 97.32 | 99.47 | |
| **HAR** | | | | | |
| PCA | 87.95 | 96.13 | 95.55 | 95.55 | **0.37** |
| Original | 87.85 | 96.10 | 96.23 | 96.47 | |
| **Spambase** | | | | | |
| PCA | 91.869 | 91.411 | 86.02 | 93.87 | **1.24** |
| Original | 92.725 | 89.58 | 91.15 | 94.72 | |
| **Isolet** | | | | | |
| PCA | 88.775 | 96.087 | 96.22 | 96.99 | **-0.13** |
| Original | 88.582 | 96.087 | 96.02 | 96.85 | |

Therefore, we first normalize the distances ($d/diameter$ and $d''/diameter$) and then plot $f_1$ and $f_2$ as a function of $\lambda$ for different datasets. Note that for PCA, we only care about the value of $f_1$ since projection along half-spaces reduces the distance between points and hence $f_2$ is always zero. For the second part, we ran four classification algorithms namely, Nearest Neighbor, Regression, Linear SVM and Gaussian SVM on the reduced as well as the original representations. We then calculated the test accuracy difference between the original and reduced representation averaged over these four classification algorithms. This we call the 'performance gap'. Our experiments show that there is some correlation between the $f_1, f_2$ values and the performance gap.

The results of our experiments are summarized in Figure 7.1 and Table 7.1. Figure 7.1 shows the plot of $f_1$ for the five different datasets. Notice that the values of $f_1$ are small ($< 0.15$) for four of the five datasets. Also, observe that for three of the datasets the $f_1$ values are very close to zero for small values of $\lambda$. In fact, for Isolet and MNIST the values are very close to zero even for very large values of $\lambda$. This shows that the PCA representations for these datasets retains distance. Even for the dataset Gassensor, the $f_1$ value is bounded by 0.25. Figure 7.1 can also be viewed as capturing a distance profile of these representations. Such a plot should be useful for other dimensionality reduction techniques as well. It gives an indication of how the reduction technique changes the distances between points.

We ran the PCA representations on different classification algorithms (Nearest Neighbor, Regression, Linear SVM and Gaussian SVM). For each of these algorithms we obtained some test accuracy. We compared these accuracies with accuracy obtained when we use the original representation. These values are listed in Table 7.1. In addition, we also calculate the accuracy difference averaged over all the classification algorithms. Notice the correlation between the accuracy gap of the datasets and the $f_1$ values. For Gassenor, the accuracy gap is the highest 2.9% and so is its $f_1$ value which goes upto 0.25. For spambase, the accuracy gap is less 1.24 and so is its $f_1$ values. For MNIST and Isolet the $f_1$ values are very very close to zero. Their accuracy gap is also very close to 0. In fact for Isolet its $-0.13$. A negative indicates that the performance of Isolet representations are in fact better than the original representation.

From our experiments in this section, we make the following observation. Datasets on which PCA retains distances in a better way tend to perform better on subsequent classification algorithms as well. In future, we would like to extend this approach to other dimensionality reduction techniques as well and see if similar conclusions can be made for them or not.

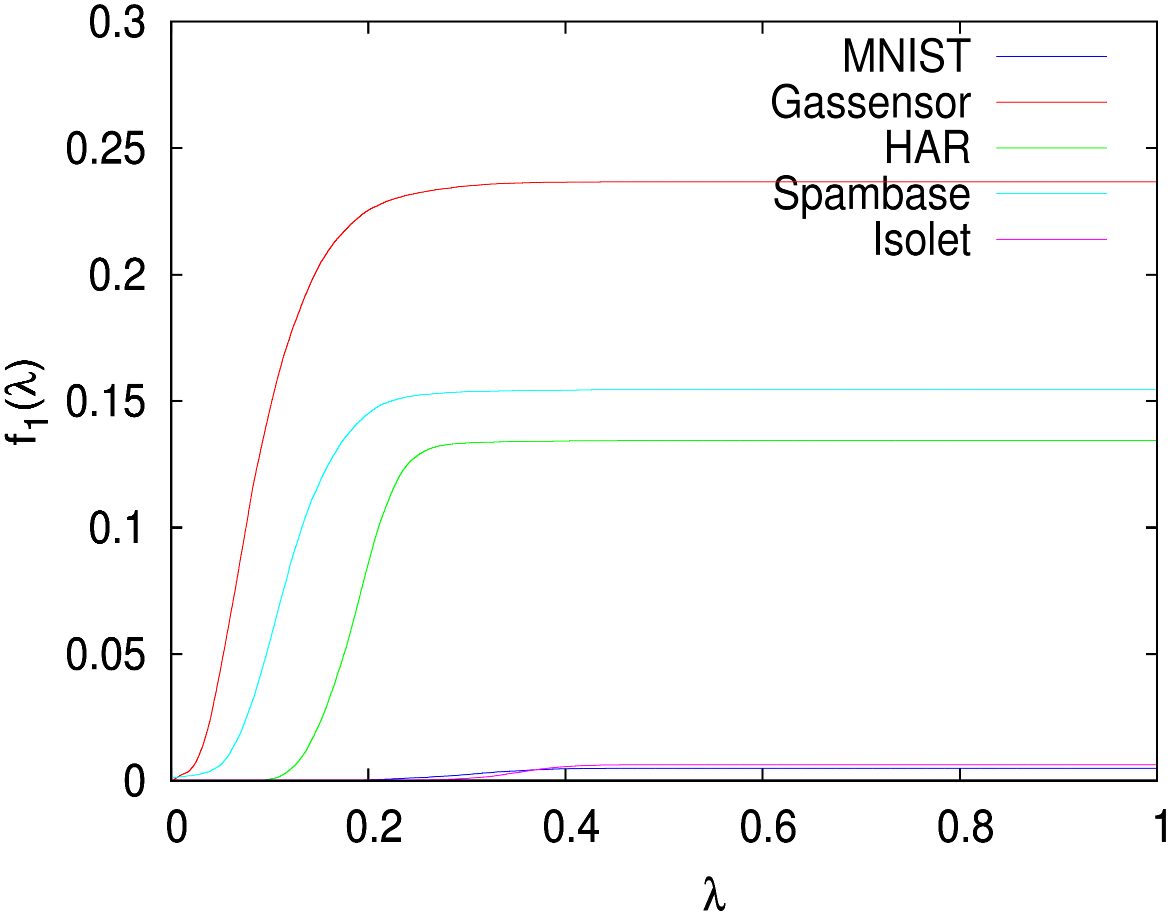

Figure 7.1: Fraction of points whose distance have shrunk by a factor of 2 plotted against their normalized distance in reduced representation.

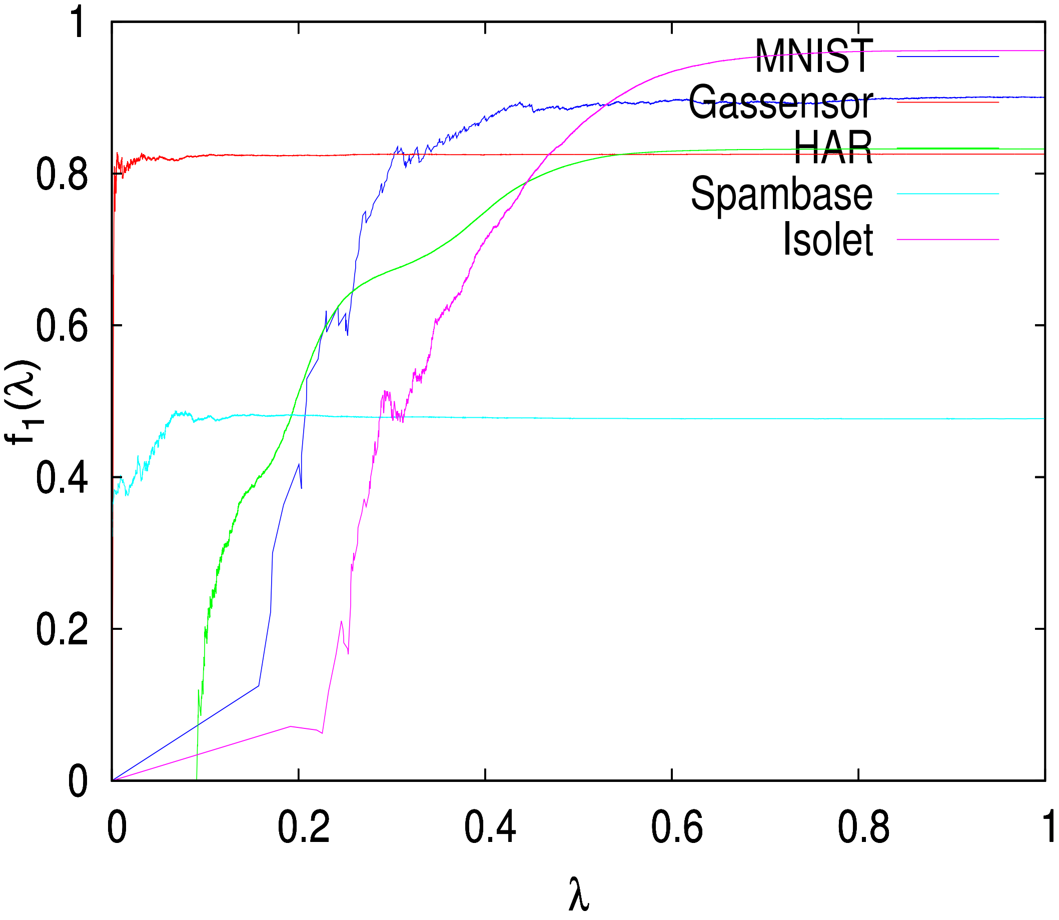

Figure 7.2: Fraction of pairs of points which have different labels given that the normalized distances are less than $\lambda$.

## 7.2 Probabilistic Lipschitzness

In the second set of experiments, we compare the performance of the two notions of Probabilistic Lipschitzness, namely PL-Unary and PL-Conditional. To evaluate the two quantities for a particular dataset, we need to calculate the probabilities introduced in Definition 4.2 and 4.1. However, calculation of these probabilities needs knowledge of the distribution which is seldom available in real-world applications. Hence, we estimate these probabilities using the following quantities.

$$f_1(\lambda) = \text{Fraction of pairs of points}\,[\,l(x) \neq l(y)\,|\,d(x,y) < \lambda\,] \;\text{ and}$$
$$f_2(\lambda) = \text{Fraction of points}\,[\,\exists y \,:\, l(x) \neq l(y) \wedge d(x,y) < \lambda\,]$$

Just as in Section 7.1, we use the normalized version of the distance function $d$. Figures 7.2 and 7.3 show the plot of $f_1$ and $f_2$ respectively for different datasets. To calculate $f_1$, we first randomly sample $100,000$ pairs of points. For each of the pairs of points we check if their labels are the same and calculate $f_1$. The calculations for $f_2$ are also somewhat similar.

Now, note that the general trend in the graph of both these functions is somewhat similar. $f_2$ is smooth increasing function of distance $\lambda$ while the graph of $f_1$ is a little bit more noisy. This is to be expected from the definitions of PL-Unary and PL-Conditional. PL-Unary is an increasing function of the parameter $\lambda$ while PL-Conditional is not. But even then, the general trend for $f_1$ is that roughly it increases with increasing value of $\lambda$. Another point to note is that for large values of $\lambda$, $f_1$ is always less than $f_2$.

Note that for each of the datasets, the values of $f_1$ and $f_2$ spike at about the same $\lambda$. Note that for Probabilistic Lipschitzness, we only care about small values of $\lambda$ in the range of two to four. We would expect points which are separated by a large distance to have different labels. Thus, our experiments suggest that both the notions of Probabilistic Lipschitzness are roughly the same in practice.

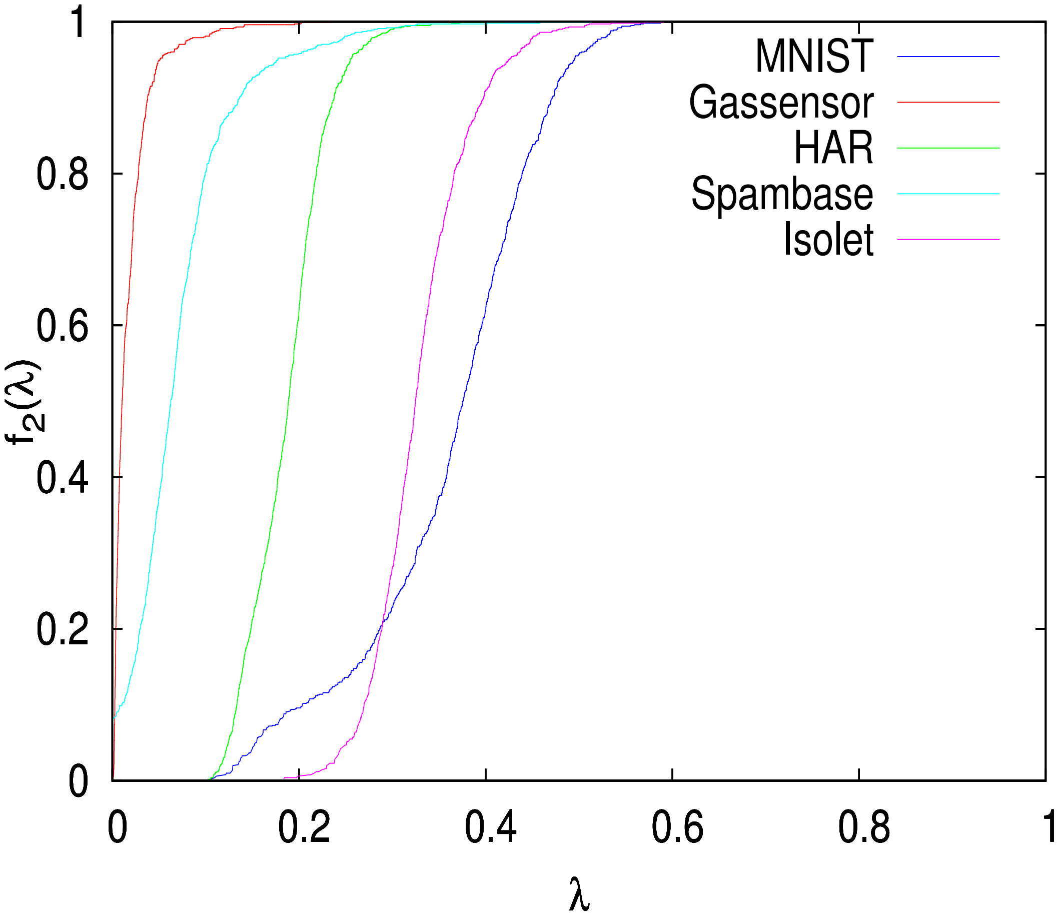

Figure 7.3: Fraction of points which have a point of different label when the normalized distance is less than $\lambda$.

# Chapter 8

# Conclusion and Future Work

In this thesis, we introduced a new notion to model dimensionality reduction techniques (or embeddings) which preserve inter-point distances called *Retaining Distance* (Def. 3.1). We showed that common techniques like PCA do have this property (Theorem 3.2). We then showed that an embedding which retains distance not only preserves inter-point distances but also some of the other 'niceness' properties of the distribution as well (Theorem 5.1). The niceness property that we considered was Probabilistic Lipschitzness (Def. 4.2). 'Close points tend to have same labels' - we showed that the notion of PL-Conditional is useful for modeling this assumption (Fig 7.2). We proved sample complexity bounds for Nearest Neighbor under PL-Conditional (Theorem 4.1). Hence, we showed that a distance retaining embedding can be used to achieve information preserving dimensionality reduction.

There are several avenues for future work. The first is extending the notion of retaining distance to other dimensionality reduction techniques. In future, we want to identify assumptions under which techniques such K-Means based reduction, Neural Autoencoders retain distance. We want to use Retaining Distance as a means to identifying assumptions under which one would expect dimensionality reduction techniques to work well. Another direction that we want to explore is using PL-Conditional in other domains. Using PL-Unary results have been obtained for Active Learning and other semi-supervised techniques. We want to see if similar results would hold under PL-Conditional assumptions. Another direction is using PL-Conditional or some similar niceness assumptions to obtain results for half-spaces, SVMs etc. In this work, we considered niceness assumptions in the domain of dimensionality reduction and labeling functions. Another interesting idea would be to consider similar assumptions in the field of Kernel Learning.

# APPENDICES

# Appendix A

# Lemmas and Proofs

For the proof of Theorem 4.1, we need a few Mathematical tools. The first is a lemma which bounds the probability of a point being far away from all the points of the sample $S$. Then we would also need Markov's inequality which upper bounds the probability that a non-negative random variable is greater than some positive constant.

**Lemma A.1.** *Let $C_1, C_2, \ldots, C_r$ be subsets of some domain set $\mathbb{X}$ and let $S \subseteq \mathbb{X}$ be an unlabeled set of size $m$ generated i.i.d by some distribution $P$ over $\mathbb{X}$. Then*

$$\underset{S \sim P^m}{\mathbb{E}\mathrm{xp}} \left[ \sum_{i : C_i \cap S = \phi} \underset{x \sim P}{\mathbb{P}\mathrm{r}} \Big[ x \in C_i \Big] \right] \leq \frac{r}{me}$$

*Proof.* The proof of this Lemma can be found in the book [26] Lemma 19.2. □

**Lemma A.2.** *Let $C_1, C_2, \ldots, C_r$ be subsets of some domain set $\mathbb{X}$ and let $S \subseteq \mathbb{X}$ be an unlabeled set of size $m$ generated i.i.d by some distribution $P$ over $\mathbb{X}$. Then for every $k \geq 2$*

$$\underset{S \sim P^m}{\mathbb{E}\mathrm{xp}} \left[ \sum_{i : |C_i \cap S| < k} \underset{x \sim P}{\mathbb{P}\mathrm{r}} \Big[ x \in C_i \Big] \right] \leq \frac{2rk}{m}$$

*Proof.* The proof of this Lemma can be found in the book [26] Lemma 19.6. □

**Lemma A.3.** *Let $x, x_1, \ldots, x_k$ be $k+1$ points generated i.i.d in $[0,1]^N$ which are labeled by a 0-1 function $l$ which satisfies PL-Conditional with function $\phi$. Assume that the points are such that $d(x, x_i) < \lambda$ for all $1 \leq i \leq k$. Let the set $A = \{x_i : l(x) \neq l(x_i)\}$. Then*

$$\mathbb{P}\mathrm{r} \Big[ |A| > t \Big] \leq \frac{k.\phi(\lambda)}{t}$$

*Proof.* $d(x, x_i) < \lambda$ and hence using PL-Conditional, we get that $\Pr[\, l(x) \neq l(x_i)] \leq \phi(\lambda)$. We will model this by a binomial distribution. Consider the event $l(x) \neq l(x_i)$ as success. Then the probability of success $p \leq \phi(\lambda)$. In this case, the event $|A| > t$ is equivalent to number of successes being greater than $t$. The expected number of successes in $k$ trials for a binomial distribution is $kp$. Using, Markov's inequality

$\Pr\ \big[\,|A| > t\big] = \Pr\big[\text{no success} > t\big] \leq \frac{\mathrm{Exp}\big[\,|A|>t\big]}{t} \leq \frac{k.\phi(\lambda)}{t}$ □

**Markov's inequality** Let $x$ be any random variable which takes only non-negative values. Let $a > 0$ be any real number and $\mu$ the expected value of the random variable $x$, then

$$\Pr_{x\sim P}\ \big[\, x \geq a \,\big] \ \leq\ \frac{\mu}{a} \tag{A.1}$$

Note that Markov's inequality is actually a more general version of the Chebyshev's inequality which we used in Section 3.2. Chebyshev's inequality can actually be derived as a corollary to the Markov inequality. We will now give the detailed proof of Theorem 4.1. The proof will use the ideas already discussed in Section 4.2.

### Proof of Theorem 4.1

We need to upper bound the quantity $\Pr_{S\sim P^m}\ [\,Err_P(NN(S)) > \epsilon\,]$. Let $S := \{x_1, \ldots, x_m\}$ be the unlabeled dataset. Partition the domain $\mathbb{X} = [0,1]^N$ into $r = (\sqrt{N}/\lambda)^N$ axis-aligned boxes $C_1, \ldots, C_r$ each of length $\lambda/\sqrt{N}$ and diameter $\lambda$ for some $\lambda$ (to be chosen later). For any $x \in [0,1]^N$ denote by $C(x)$ the region in which $x$ lies and by $\pi_S(x)$ the Nearest Neighbor of $x$ in $S$.

$$\begin{aligned}
&Err_P(NN(S)) = \Pr_{x\sim P}\ [\, l(\pi_S(x)) \neq l(x)\,]\\
&= \Pr_{x\sim P}\ [\, C(x)\cap S = \phi \wedge l(\pi_S(x)) \neq l(x)\,] + \Pr_{x\sim P}\ [\, C(x)\cap S \neq \phi \wedge l(\pi_S(x)) \neq l(x)\,]\\
&\leq \Pr_{x\sim P}\ [\, C(x)\cap S = \phi\,] + \Pr_{x\sim P}\ [\, C(x)\cap S \neq \phi \wedge l(\pi_S(x)) \neq l(x)\,]\\
&= \sum_{i:C_i\cap S=\phi} \Pr_{x\sim P}\ [\, x\in C_i\,] + \sum_{i:C_i\cap S\neq\phi} \Pr_{x\sim P}\ [\, x \in C_i \wedge l(\pi_S(x)) \neq l(x)\,]
\end{aligned}$$

For notational convenience, denote $\mathbb{P}[C_i] := \Pr_{x\sim P}\ [\, x \in C_i\,]$. Then the above becomes

$$= \sum_{i:C_i\cap S=\phi} \mathbb{P}[C_i] + \sum_{i:C_i\cap S\neq\phi} \mathbb{P}[C_i] \Pr_{x\sim P_{C_i}}\ [\, l(\pi_S(x)) \neq l(x)\,]$$

where $P_{C_i}$ denotes the distribution $P$ restricted to the set $C_i$. Observe that, since $C_i \cap S \neq \phi$, we have that $d(x, \pi_S(x)) \leq \lambda$.

$$\therefore Err_P(NN(S)) \leq \sum_{i:C_i \cap S = \phi} \mathbb{P}[C_i] + \sum_{i:C_i \cap S \neq \phi} \mathbb{P}[C_i] \underset{x \sim P_{C_i}}{\mathbb{Pr}} \left[ l(\pi_S(x)) \neq l(x) \right] \tag{A.2}$$

For brevity of notation, denote $a(x,y) := (l(x) \neq l(y))$. Now consider the expectation over the sample S of the quantity on the extreme right of Equation A.2.

$$\underset{S \sim P^m}{\mathbb{Exp}} \left[ \sum_{i:C_i \cap S \neq \phi} \mathbb{P}[C_i] \underset{x \sim P_{C_i}}{\mathbb{Pr}} \left[ a(\pi_S(x), x) \right] \right] = \sum_{i=1}^{r} \mathbb{P}[C_i] \underset{S \sim P^m}{\mathbb{Exp}} \left[ \underset{x \sim P_{C_i}}{\mathbb{Pr}} \left[ a(\pi_S(x), x) \right] \mathbf{1}_{C_i \cap S \neq \phi} \right]$$

$$\leq \sum_{i=1}^{r} \mathbb{P}[C_i] \underset{S \sim P^m}{\mathbb{Exp}} \left[ \underset{x \sim P_{C_i}}{\mathbb{Pr}} \left[ a(\pi_S(x), x) \right] \right]$$

$$= \sum_{i=1}^{r} \mathbb{P}[C_i] \underset{S \sim P^m}{\mathbb{Exp}} \left[ \underset{x \sim P_{C_i}}{\mathbb{Exp}} \left[ \mathbf{1}_{a(\pi_S(x),x)} \right] \right] \quad = \quad \sum_{i=1}^{r} \mathbb{P}[C_i] \underset{\substack{S \sim P^m \\ x \sim P_{C_i}}}{\mathbb{Exp}} \left[ \mathbf{1}_{a(\pi_S(x),x)} \right]$$

$$= \sum_{i=1}^{r} \mathbb{P}[C_i] \underset{\substack{S \sim P^m \\ x \sim P_{C_i}}}{\mathbb{Pr}} \left[ a(\pi_S(x), x) \right] \tag{A.3}$$

Now, consider the quantity $\mathbb{P}[a(\pi_S(x), x)]$ in Equation A.3. Using conditional probability, we get that

$$\underset{\substack{S \sim P^m \\ x \sim P_{C_i}}}{\mathbb{Pr}} \left[ a(\pi_S(x), x) \right] = \sum_{j=1}^{m} \underset{\substack{S \sim P^m \\ x \sim P_{C_i}}}{\mathbb{Pr}} \left[ a(\pi_S(x), x) \,|\, \pi_S(x) = x_j \right] \underset{\substack{S \sim P^m \\ x \sim P_{C_i}}}{\mathbb{Pr}} \left[ \pi_S(x) = x_j \right]$$

Denote $\mathbb{P}_S[\pi_{x_j}] := \underset{\substack{S \sim P^m \\ x \sim P_{C_i}}}{\mathbb{Pr}} [\pi_S(x) = x_j]$. Now, observe that the labeling function $l$ is independent of the choice of the nearest Neighbor $\pi_S(x)$ which depends only on the distance function $d$. Hence, we get

$$\underset{\substack{S \sim P^m \\ x \sim P_{C_i}}}{\mathbb{Pr}} \left[ a(\pi_S(x), x) \right] \quad = \quad \sum_{j=1}^{m} \mathbb{P}_S[\pi_{x_j}] \underset{\substack{S \sim P^m \\ x \sim P_{C_i}}}{\mathbb{Pr}} \left[ a(x_j, x) \right]$$

$$\leq \sum_{j=1}^{m} \mathbb{P}_S[\pi_{x_j}] \underset{\substack{x_1, \ldots, x_m \sim P \\ x \sim P_{C_i}}}{\mathbb{Pr}} \left[ a(x_j, x) \right]$$

$$= \sum_{j=1}^{m} \mathbb{P}_S[\pi_{x_j}] \underset{\substack{x_j \sim P \\ x \sim P_{C_i}}}{\mathbb{Pr}} \left[ l(x_j) \neq l(x) \right]$$

Now using the fact that the labeling function satisfies $\phi$-PL, we get that the above quantity is upper bounded by

$\leq \sum_{j=1}^{m} \mathbb{P}_S[\pi_{x_j}]\,\phi(\lambda) \;\leq\; \phi(\lambda)$. Substituting this in Eqn A.3, we get

$$\underset{S\sim P^m}{\mathbb{E}\text{xp}}\Big[\sum_{i:C_i\cap S\neq\phi}\mathbb{P}[C_i]\underset{x\sim P_{C_i}}{\mathbb{P}\text{r}}\,[\,NN(S)(x)\neq l(x)\,]\Big] \;\leq\; \sum_{i=1}^{r}\mathbb{P}[C_i]\;\phi(\lambda) \;\leq\; \phi(\lambda) \tag{A.4}$$

Now, using Equations A.2 and A.4 together with Lemma A.1, we get that

$$\underset{S\sim P^m}{\mathbb{E}\text{xp}}\big[\,Err_P(NN(S))\,\big] \;\leq\; \frac{r}{me}+\phi(\lambda)$$

Using Markov's inequality (Equation A.1) and substituting values we get

$$\underset{S\sim P^m}{\mathbb{P}\text{r}}\big[\,Err_P(NN(S))>\epsilon\,\big] \;\leq\; \frac{1}{me\epsilon}\left(\frac{\sqrt{N}}{\lambda}\right)^{N}+\frac{\phi(\lambda)}{\epsilon}$$

Using $\lambda=\phi^{-1}(\epsilon\delta/2)$, we get the result of the Theorem. □

**Proof of Theorem 4.2**

The proof is very similar to the proof of theorem 4.1. However, for completeness, we state this proof in complete detail. We need to upper bound the quantity $\underset{S\sim P^m}{\mathbb{P}\text{r}}\,[\,Err_P(kNN(S))>\epsilon\,]$. Let $S:=\{x_1,\ldots,x_m\}$ be the unlabeled dataset. Partition the domain $\mathbb{X}=[0,1]^N$ into $r=(\sqrt{N}/\lambda)^N$ axis-aligned boxes $C_1,\ldots,C_r$ each of length $\lambda/\sqrt{N}$ and diameter $\lambda$ for some $\lambda$ (to be chosen later). For any $x\in[0,1]^N$ denote by $C(x)$ the region in which $x$ lies and by $\pi_1(x),\ldots,\pi_k(x)$ the $k$ nearest neighbors of $x$ in $S$. Let $A_{\pi_1,\ldots,\pi_k}(x):=\{\pi_i(x):l(x)\neq l(\pi_i(x))\}$. That is, $A_{\pi_1,\ldots,\pi_k}(x)$ is the set of neighbors of $x$ which have different labels than $x$. Assuming that the labeling function $l$ is $\{0,1\}$, we get that

$$Err_P(kNN(S))=\underset{x\sim P}{\mathbb{P}\text{r}}\;[\;|\,A_{\pi_1,\ldots,\pi_k}(x)\,|\;>k/2\,]$$

Now, we have two possibilities. Either all the $k$ nearest neighbors are such that $d(x,\pi_i(x))<\lambda$ or there exists atleast one $i$ such that $d(x,\pi_i(x))>\lambda$. In the latter case, we have that $|C(x)\cap S|<k$. Hence, we get that the error above

$$\leq \underset{x\sim P}{\mathbb{P}\text{r}}\;[\;|\,C(x)\cap S\,|<k\,]+\underset{x\sim P}{\mathbb{P}\text{r}}\;[\;|\,A_{\pi_1,\ldots,\pi_k}(x)\,|>k/2\;\mid\;d(x,\pi_i(x))<\lambda\text{ for all }i]$$

$$\therefore Err_P(kNN(S))\leq \sum_{i:|C_i\cap S|<k}\mathbb{P}[C_i]\;+\underset{x\sim P}{\mathbb{P}\text{r}}[\;|\,A_{\pi_1,\ldots,\pi_k}(x)\,|>k/2\,] \tag{A.5}$$

where $\mathbb{P}[C_i] := \underset{x \sim P}{\mathbb{Pr}}\,[\,x \in C_i\,]$. Now consider the expectation over the sample S of the quantity on the right hand size of Equation A.5. Note that in this case, for all $i = 1$ to $k$, we have that $d(x, \pi_i(x)) < \lambda$. We omit this in equation A.5 only for brevity of notation.

$$\underset{S \sim P^m}{\mathbb{Exp}}\left[\underset{x \sim P}{\mathbb{Pr}}\,[\,|\,A_S(x)\,| > k/2\,]\right] = \underset{S \sim P^m}{\mathbb{Exp}}\left[\underset{x \sim P}{\mathbb{Exp}}\left[\,\mathbf{1}_{|A_{\pi_1,\ldots,\pi_k}(x)| > k/2}\,\right]\right] = \underset{\substack{S \sim P^m \\ x \sim P}}{\mathbb{Exp}}\left[\,\mathbf{1}_{|A_{\pi_1,\ldots,\pi_k}(x)| > k/2}\right]$$
$$= \underset{\substack{S \sim P^m \\ x \sim P}}{\mathbb{Pr}}\left[\,|A_{\pi_1,\ldots,\pi_k}(x)| > k/2\right] \tag{A.6}$$

Using conditional probability over the choice of $k$ nearest neighbors, we get that the above

$$= \sum_{\text{all k tuples}} \underset{\substack{S \sim P^m \\ x \sim P}}{\mathbb{Pr}}\left[\,|\,A_{x_{i_1},\ldots,x_{i_k}}(x)\,| > k/2 | \pi_1 = x_{i_1}, \ldots, \pi_k = x_{i_k}\right] \underset{\substack{S \sim P^m \\ x \sim P}}{\mathbb{Pr}}[\,\pi_1 = x_{i_1}, \ldots, \pi_k = x_{i_k}\,]$$

Denote by $\mathbb{P}_S[x_{i_1}, \ldots, x_{i_k}]$ the quantity on the extreme right of above equation. Also, observe that $|A_{x_{i_1},\ldots,x_{i_k}}(x)|$ depends only on the labeling function $l$ which is independent of the choice of the nearest neighbors. Hence, we get

$$\underset{\substack{S \sim P^m \\ x \sim P}}{\mathbb{Pr}}\left[\,|A_{\pi_1,\ldots,\pi_k}(x)| > k/2\right] = \sum_{\text{all k tuples}} \mathbb{P}_S[x_{i_1}, \ldots, x_{i_k}] \underset{\substack{S \sim P^m \\ x \sim P}}{\mathbb{Pr}}\,[\,|A_{x_{i_1},\ldots,x_{i_k}}(x)| > k/2]$$
$$\leq \sum_{\text{all k tuples}} \mathbb{P}_S[x_{i_1}, \ldots, x_{i_k}]\; 2\phi(\lambda) \;\; \text{(Using Lemma A.3)}$$

$\leq 2\,\phi(\lambda)$. Substituting this in Eqn A.6, we get

$$\underset{S \sim P^m}{\mathbb{Exp}}\left[\underset{x \sim P}{\mathbb{Pr}}\,[\,|\,A_S(x)\,| > k/2\,]\right] \;\leq\; 2\phi(\lambda) \tag{A.7}$$

Now, using Equations A.5 and A.7 together with Lemma A.2, we get that

$$\underset{S \sim P^m}{\mathbb{Exp}}\left[\,Err_P(NN(S))\,\right] \;\leq\; \frac{2rk}{m} + 2\phi(\lambda)$$

Using Markov's inequality (Equation A.1) and substituting values we get

$$\underset{S \sim P^m}{\mathbb{Pr}}\left[\,Err_P(kNN(S)) > \epsilon\,\right] \;\leq\; \frac{2k}{m\epsilon}\left(\frac{\sqrt{N}}{\lambda}\right)^N + \frac{2\phi(\lambda)}{\epsilon}$$

Using $\lambda = \phi^{-1}(\epsilon\delta/4)$, we get the result of the Theorem. □

**Proof of Theorem 4.3**

As in the proof of Theorem 4.2, let $S := \{x_1, \ldots, x_m\}$ be the unlabeled dataset. Partition the domain $\mathbb{X} = [0,1]^N$ into $r = (\sqrt{N}/\lambda)^N$ axis-aligned boxes $C_1, \ldots, C_r$ each of length $\lambda/\sqrt{N}$ and diameter $\lambda$. For any $x \in [0,1]^N$ denote by $C(x)$ the region in which $x$ lies and by $\pi_1(x), \ldots, \pi_k(x)$ the $k$ nearest neighbors of $x$ in $S$.

Now, there are two possibilities. In the first case assume that there exists atleast one $i$ such that $d(x, \pi_i(x)) > \lambda$. In this case, $|C(x) \cap S| < k$ and we can use Lemma A.2 to bound the probability of error. In the other case we have that, for all $i$, $d(x, \pi_i(x)) < \lambda$. In this case, the error is upper bounded by the event that there exists $k/2$ points of label different than $x$. Hence, we get that

$$Err_P(kNN(S)) \leq \sum_{i:|C_i \cap S| < k} \mathbb{P}[C_i] + \Pr_{x \sim P}\big[\exists y_1, \ldots, y_{k/2} : \text{ for all i } l(x) \neq l(y_i) \wedge d(x, y_i) < \lambda\big]$$

$$\leq \sum_{i:|C_i \cap S| < k} \mathbb{P}[C_i] + \phi(\lambda)^{k/2} \tag{A.8}$$

where $\mathbb{P}[C_i] := \Pr_{x \sim P}[\, x \in C_i \,]$. Now, using Lemma A.2, we get that

$$\mathbb{E}\mathrm{xp}_{S \sim P^m}\big[\, Err_P(kNN(S)) \,\big] \;\leq\; \frac{2rk}{m} + \phi(\lambda)^{k/2}$$

Using Markov's inequality (Equation A.1) and substituting values we get

$$\Pr_{S \sim P^m}\big[\, Err_P(kNN(S)) > \epsilon \,\big] \;\leq\; \frac{2k}{m\epsilon}\left(\frac{\sqrt{N}}{\lambda}\right)^N + \frac{\phi(\lambda)^{k/2}}{\epsilon}$$

Using $\lambda = \phi^{-1}((\epsilon\delta/2)^{2/k})$, we get the result of the Theorem. □

# References

[1] Ittai Abraham, Yair Bartal, and Ofer Neiman. Embedding metric spaces in their intrinsic dimension. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 363–372. Society for Industrial and Applied Mathematics, 2008.

[2] Noga Alon. Problems and results in extremal combinatoricsi. *Discrete Mathematics*, 273(1):31–53, 2003.

[3] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *Ambient Assisted Living and Home Care*, pages 216–223. Springer, 2012.

[4] David Arthur, Bodo Manthey, and H Roglin. k-means has polynomial smoothed complexity. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 405–414. IEEE, 2009.

[5] David Arthur and Sergei Vassilvitskii. How slow is the k-means method? In *Proceedings of the twenty-second annual symposium on Computational geometry*, pages 144–153. ACM, 2006.

[6] Sunil Arya, David M Mount, Nathan S Netanyahu, Ruth Silverman, and Angela Y Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM)*, 45(6):891–923, 1998.

[7] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[8] Yair Bartal, Ben Recht, and Leonard J Schulman. Dimensionality reduction: beyond the johnson-lindenstrauss bound. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 868–887. SIAM, 2011.

[9] Richard Ernest Bellman. Some new techniques in the dynamic-programming solution of variational problems. 1957.

[10] Richard Ernest Bellman. *Adaptive control processes: a guided tour*, volume 4. Princeton university press Princeton, 1961.

[11] Shai Ben-David, Tyler Lu, and Dávid Pál. Does unlabeled data provably help? worst-case analysis of the sample complexity of semi-supervised learning. In *COLT*, pages 33–44, 2008.

[12] Avrim L Blum and Ronald L Rivest. Training a 3-node neural network is np-complete. *Neural Networks*, 5(1):117–127, 1992.

[13] Adam Coates and Andrew Y Ng. Learning feature representations with k-means. In *Neural Networks: Tricks of the Trade*, pages 561–580. Springer, 2012.

[14] Adam Coates, Andrew Y Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics*, pages 215–223, 2011.

[15] Ali Ghodsi. Dimensionality reduction a short tutorial. *Department of Statistics and Actuarial Science, Univ. of Waterloo, Ontario, Canada*, 2006.

[16] Lee-Ad Gottlieb and Robert Krauthgamer. A nonlinear approach to dimension reduction. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 888–899. SIAM, 2011.

[17] Anupam Gupta, Robert Krauthgamer, and James R Lee. Bounded geometries, fractals, and low-distortion embeddings. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, pages 534–543. IEEE, 2003.

[18] Venkatesan Guruswami and Prasad Raghavendra. Hardness of learning halfspaces with noise. *SIAM Journal on Computing*, 39(2):742–765, 2009.

[19] Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.

[20] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[21] Laurent Hyafil and Ronald L Rivest. Constructing optimal binary decision trees is np-complete. *Information Processing Letters*, 5(1):15–17, 1976.

[22] Mary Inaba, Naoki Katoh, and Hiroshi Imai. Applications of weighted voronoi diagrams and randomization to variance-based k-clustering. In *Proceedings of the tenth annual symposium on Computational geometry*, pages 332–339. ACM, 1994.

[23] William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

[24] Donald E Knuth. The art of computer programming: Fundamental algorithms, vol. i, 1968.

[25] Maxim Raginsky and Alexander Rakhlin. Lower bounds for passive and active learning. In *Advances in Neural Information Processing Systems*, pages 1026–1034, 2011.

[26] Shai Shalev-Shwartz and Shai Ben-David. Understanding machine learning. 2014.

[27] Ingo Steinwart and Clint Scovel. Fast rates for support vector machines. In *Learning Theory*, pages 279–294. Springer, 2005.

[28] Ruth Urner and Shai Ben-David. Probabilistic lipschitzness a niceness assumption for deterministic labels. In *Learning Faster from Easy Data-Workshop@ NIPS*, 2013.

[29] Ruth Urner, Shai Ben-David, and Shai Shalev-Shwartz. Access to unlabeled data can speed up prediction time. In *ICML*, 2011.

[30] Laurens JP van der Maaten, Eric O Postma, and H Jaap van den Herik. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*, 10(1-41):66–71, 2009.

[31] Alexander Vergara, Shankar Vembu, Tuba Ayhan, Margaret A Ryan, Margie L Homer, and Ramón Huerta. Chemical gas sensor drift compensation using classifier ensembles. *Sensors and Actuators B: Chemical*, 166:320–329, 2012.

[32] Roger Weber, Hans-Jörg Schek, and Stephen Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *VLDB*, volume 98, pages 194–205, 1998.