8-5-2010

On Dimensionality Reduction of Data

Harika Rao Vamulapalli
University of New Orleans

On Dimensionality Reduction of Data

A Thesis

Submitted to the Graduate Faculty of the
University of New Orleans
in partial fulfillment of the
requirements for the degree of

Master of Science
in
Engineering

by

Harika Rao Vamulapalli

B.Tech. Jawaharlal Nehru Technological University, 2007

August, 2010

Acknowledgements

I wish to thank those people who contributed to this thesis both directly and indirectly. Firstly, I would like to express my gratitude to my supervisor, Dr. Huimin Chen, whose encouragement and support in every aspect to reach the goals, really helped me all the time of my research. I appreciate his patience, guidance and supervision of my work which helped me in the right path. I would like to thank other members in my thesis committee, Dr. X. Rong Li and Dr. Vesselin P. Jilkov, for their support. I would like to thank my labmate, Mr. Gang Liu who have contributed immensely to my personal and professional time. In addition, I wish to dedicate this work to my family and friends who encouraged and supported me during all this time.

Finally, I am grateful to the financial support in part by the research grants from Army Research Office (ARO W911NF-08-1-0409), Louisiana Board of Regents (NSF-2009-PFUND-162), Office of Naval Research (DEPSCoR N00014-09-1-1169) and NASA (EPSCOR DART2).

Table of Contents

List of Figures

List of Tables

Abstract

Random projection method is one of the important tools for the dimensionality reduction of data which can be made efficient with strong error guarantees. In this thesis, we focus on linear transforms of high dimensional data to the low dimensional space satisfying the Johnson-Lindenstrauss lemma. In addition, we also prove some theoretical results relating to the projections that are of interest when applying them in practical applications. We show how the technique can be applied to synthetic data with probabilistic guarantee on the pairwise distance. The connection between dimensionality reduction and compressed sensing is also discussed.

Keywords: principal component analysis, dimensionality reduction, random projection, fast Johnson Lindenstrauss transform

Chapter 1

Introduction

1.1 Motivations & Background

In many machine learning problems, the data available to us can be viewed as distinct objects or items, each of which has a number of attributes. For the sake of clustering or classifying data into different groups, one may model each data entry as a point in certain high-dimensional space. For n data points, it is convenient to represent the collection of data points by a matrix

$$A = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \cdots \\ \mathbf{u}_n \end{bmatrix}$$

where each data point $\mathbf{u}_i$ is assumed to have d dimensions.

The problem posed by high-dimensional data is trivial to state, but not so simple to solve. Many algorithms of practical interest such as clustering and classification algorithms can not handle a large number of dimensions efficiently. In addition, a deeper issue called the "curse of dimensionality" has plagued researchers in machine learning and other fields for decades. Generally speaking, for any data entry, as we increase the number of dimensions that it possesses, the complexity involved in processing it for inference purposes increases at an exponential rate. This poses a dilemma: naturally, we want to use as much information as is available to us, on the other hand, as we collect more information, we have to spend dramatically more time trying to make sense of it. Fortunately, there has been abundant research works trying to get around the problems of dealing with data of high-dimensionality. Most of the techniques can be seen as to approximate the high-dimensional data with low-dimensional representations and at the same time to maintain the key structure of the original data to the best extent. We call these techniques a method of *dimensionality reduction*. Formally, dimensionality reduction involves a mapping from a high-dimensional space to a lower-dimensional one where certain "distance" concept can be preserved.

Note that the distance concept between any pair of data points depends on the nature of problem. The Euclidean distance may not be meaningful for binary matrix, which could represent the positive and negative instances of certain words for document categorization. In such a problem, the Hamming distance seems to be an appropriate measure. Note also

that the data with reduced dimensions can not be viewed as the compressed version of the original data since data compression concerns to recover the original data from their compressed representations with a distortion below certain desired level. The effectiveness of data compression is usually measured in terms of the number of bits used to represent the data instead of the dimensionality reduction. On the other hand, dimensionality reduction cares about the pairwise distance between two data points rather than the data point itself, which sometimes can be summarized by its distance to the origin.

1.1.1 Research Objective

In machine learning field, the constantly growing data dimension causes severe problems. This difficulty is a problem of long standing and continuing effort. One way is to reduce the dimension: to map the original data in high dimension to another space of low dimension, while preserving important properties as much as possible. There are many ways that map the higher dimensional data to lower dimensional manifold. We would like to focus on methods that map the higher dimensional data into some lower dimensional manifold without distorting the pairwise distances. This thesis studies the properties of random projection methods which provide a guarantee of preserving pairwise distances in the lower dimensional space with small distortion. We perform experimental study of two random projection algorithms and compare their performance to the theoretical limit delineated by the Johnson-Lindenstrauss lemma.

1.1.2 Organization of Thesis

The rest of the thesis is organized as follows. Chapter 2 provides background on dimensionality reduction via random projection and its connection to compressed sensing. Chapter 3 contains the analysis of the random projection algorithms. Chapter 4 provides the experimental study of two random projection algorithms based on the synthetic data. Chapter 5 concludes this thesis and indicates possible future directions for high dimensional data analysis.

Chapter 2

Dimensionality Reduction

2.1 Introduction

Dimensionality reduction is a method of obtaining the information from a high dimensional feature space using fewer intrinsic dimensions. In machine learning, it is very important to reduce high dimensional data set for better classification, regression, presentation and visualization of data. It is also useful for better understanding of the correlations within the data. This enables us to find the intrinsic dimensionality of the data and provide possibly better generalization capability.

2.1.1 Principal Component Analysis

Principal component analysis (PCA) is perhaps the most popularly used method in dimensionality reduction. Suppose that we want to reduce d-dimensional data to k for n-point data matrix A. Assume that $n > d$ and $\text{rank}(A) = d$, then one possible measure between original data matrix A and its k-dimensional representation B is the Frobenius norm given by

$$||A||_F = \sum_i \sum_j |a_{ij}|^2$$

and we want to find the best approximation B^* such that

$$B^* = \arg \min_{\text{rank}(B)=k} ||A - B||_F \tag{2.1}$$

Solution to the above problem boils down to the singular value decomposition (SVD) of A given by

$$A = U\Lambda V^T$$

where U and V are orthonormal matrices and D is a diagonal matrix with entries

$$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$$

called the singular values of A. If we take the first k singular values of A and collect the corresponding left and right singular vectors, then we obtain the rank-k approximation of A given by

$$A_k = U_k \Lambda_k V_k^T.$$

It can be shown [8] that $B^* = A_k$ for any $k < d$. Note that the PCA minimizes the mean square error between the original data matrix and the rank-k approximation matrix, however, the pairwise distances between the data in the original space and those in their low-rank approximations can fluctuate arbitrarily. Thus the PCA may find effective low-dimensional features from the original data set but not so meaningful in sketching/hashing high-dimensional data for clustering or classification.

2.1.2 Random Projection Methods

Consider the dimensionality reduction via linear transform so that we can obtain the reduced dimensional data presentation B via

$$B = AR$$

where R is a $d \times k$ matrix suitable for preserving the pairwise distance. Note that in Hilbert space, finding such a transform is possible thanks to the following lemma by Johnson and Lindenstrauss [9].

Lemma 1: Suppose that we have n data points in d-dimensional space. Then $\forall \epsilon > 0$, there exists a locally Lipschitz mapping $f : \mathcal{R}^d \to \mathcal{R}^k$ such that $\forall k \geq 12\frac{\log n}{\epsilon^2}$ and any two rows $\mathbf{u}, \mathbf{v} \in A$, we have

$$(1-\epsilon)||f(\mathbf{u}) - f(\mathbf{v})||^2 \leq ||\mathbf{u} - \mathbf{v}||^2 \leq (1+\epsilon)||f(\mathbf{u}) - f(\mathbf{v})||^2 \tag{2.2}$$

Note that the Johnson-Lindenstrauss lemma says that we can preserve the pairwise distance up to a distortion level within $(1 \pm \epsilon)$ if $k = O(\frac{\log n}{\epsilon^2})$. The original dimension d of each data point is irrelevant as far as the pairwise distance is concerned. If we have only two data points, then even if they have hundreds or millions of dimension, the mapping to $O(1/\epsilon^2)$ dimensional space preserves the distance within $(1 \pm \epsilon)$. In fact, for any data set A, we can always find a mapping f that preserves the pairwise distance within the desired distortion level. However, for different As, the mapping may not be the same. In fact, the proof in [9] is non-constructive – it shows the existence of f but does not say how we actually find one. In practice, there are ways to find a mapping efficiently with high probability satisfying the Johnson-Lindenstrauss lemma. These methods usually rely on random projection of the original high-dimensional data to low-dimensional subspace. By exploiting the concentration property of the pairwise distance, the probability of success can be made arbitrarily large. In fact, the authors in [1] provided an explicit construction of the mapping R summarized by the following lemma.

Lemma 2: Suppose that A is an $n \times d$ data matrix and one needs to reduce the dimension of each data point to k. Let R be a $k \times d$ matrix with entries r_{ij} given by the following

distribution:

$$r_{ij} = \begin{cases} \sqrt{3} & \text{with probability } \frac{1}{6} \\ 0 & \text{with probability } \frac{2}{3} \\ -\sqrt{3} & \text{with probability } \frac{1}{6} \end{cases}$$

Let $B = \frac{1}{\sqrt{k}}AR$. For any row $\mathbf{u}$ in A, denote by $f(\mathbf{u})$ the corresponding row in B. For any fixed $\beta > 0$ and $\epsilon > 0$, if $k > \frac{4+2\beta}{\epsilon^2/2-\epsilon^3/3}\log n$, then for any pair of distinct rows $\mathbf{u}$ and $\mathbf{v}$ in A, with probability at least $(1 - n^{-\beta})$, we have (2.2).

Note that the above lemma can be interpreted as "I will preserve all pairwise distances arbitrarily likely in the sense of (2.2) for any given ϵ if you allow me to repeatedly generate R for many times". This is one of the appealing properties that random projection methods can offer. In fact, if one wants to reduce d-dimensional data points to k-dimensional points with the distortion of pairwise distance to be within $(1 \pm \epsilon)$, most dimensionality reduction methods such as PCA can not say anything about the lowest dimension k that guarantees the mapping to be within the desired distortion level for arbitrarily chosen n data points.

2.1.3 Fast Johnson-Lindenstrauss Transform

Ailon and Chazelle proposed an efficient algorithm to compute the random projection matrix and the method is called the fast Johnson-Lindenstrauss transform (FJLT) [2]. In particular, the transform is given by

$$R = PHD \tag{2.3}$$

where the matrices P and D are random and H is deterministic. The $k \times d$ matrix P is sparse. Each entry of P is either 0 with probability $(1-q)$ or a random number independently drawn from normal distribution with zero mean and variance q^{-1} with probability q where $q = \min\left\{\Theta\left(\frac{\epsilon^{p-2}\log^p n}{d}\right), 1\right\}$. The $d \times d$ matrix H is a Walsh-Hadamard matrix where d is usually a power of 2. The $d \times d$ matrix D is a diagonal matrix with the i-th diagonal element D_{ii} drawn independently from $\{-1, 1\}$ with probability $\frac{1}{2}$.

The computation of Hx can be evaluated in $O(d \log d)$ operations by an algorithm of fast Fourier transform type [2]. However, computing $PHDx$ requires only $O(d \log k)$. Note that the Walsh-Hadamard matrix has a recursive structure as follows.

$$H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

$$H_d = \begin{pmatrix} H_{d/2} & H_{d/2} \\ H_{d/2} & -H_{d/2} \end{pmatrix}$$

Let $z = Dx$ and let z_1 and z_2 be the first and second halves of z. In addition, P_1 and P_2 are the left and right halves of P, respectively. We can compute the random projection by

$$PH_dz = \begin{pmatrix} P_1 & P_2 \end{pmatrix} \begin{pmatrix} H_{d/2} & H_{d/2} \\ H_{d/2} & -H_{d/2} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$$

$$P_1 H_{d/2}(z_1 + z_2) + P_2 H_{d/2}(z_1 - z_2)$$

Assume that P_1 and P_2 contain k_1 and k_2 nonzeros, respectively. Let $T(d,k)$ be the number of operations required to compute k coefficients out of a $d \times d$ Walsh-Hadamard transform. The recurrence relation is given by

$$T(d,k) = T(d/2, k_1) + T(d/2, k_2) + d.$$

In both cases, we have $T(d,0) = 0$ and $T(d,1) = d$. We use induction to show that $T(d,k) \leq 2d\log(k+1)$. From the above definition, we can see that the computation of the random projection should satisfy

$$\begin{aligned} T(d,k) &= T(d/2,k_1) + T(d/2,k_2) + d \\ &\leq d\log(2(k_1+1)(k_2+1)) \\ &\leq d\log((k_1+k_2+1)^2) \ \text{(for } \ k_1+k_2 = k \geq 1) \\ &\leq 2d\log(k+1) \end{aligned}$$

Thus we can see that the fast Johnson-Lindenstrauss transform only takes

$$T(d,k) = O(d\log k)$$

shown in [2].

2.2 Computational Aspects of Dimensionality Reduction

In principle, PCA finds the directions of maximal variance from the high-dimensional data and projects the data onto these directions. The runtime of PCA is $O(nd^2)$ and it will be infeasible for problems with very large dimension d. In contrast, random projection methods usually have computational complexity linear in d. In addition, since computing the distance in the original data space takes $\Omega(d)$ time while computing the distance in the reduced dimensional space takes $\Omega(k)$ time, when $k << d$, we can infer the distances among data points in the original data space by computing the distances of data points in the reduced dimensional space if the mapping satisfies Johnson-Lindenstrauss lemma. This seems to be an important aspect when one has to cluster high-dimensional data points efficiently.

The construction of R to satisfy (2.2) with high probability usually relies on randomized techniques. In [7], the authors used independent and identically distributed Gaussian random variable for r_{ij}. Recently, Ailon *et al* showed that R can be constructed in with sparse Gaussian entries such that the mapping $B = AR$ takes $O(k^3)$ runtime [2]. For k not so small compared to d, the 4-wise independent code matrix was used in [3] to improve the runtime of the mapping to $O(d\log k)$ and [5] showed that the runtime can be made in $O(d)$ with Lean Walsh transform.

2.3 Connection to Compressed Sensing

In compressed sensing, one is interested in sketching d-dimensional signal $\mathbf{x}$ by a k-dimensional measurement $\mathbf{y}$ through some encoder Φ such that given

$$\mathbf{y} = \Phi\mathbf{x}$$

one is able to recover $\mathbf{x}$ through $\mathbf{y}$ by pursuing the maximal sparsity of $\mathbf{x}$. Note that if $\mathbf{x}$ is truly a sparse signal, i.e., most of the elements in $\mathbf{x}$ are zero, then one can solve the following constrained optimization problem to recover $\mathbf{x}$.

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} ||\mathbf{x}||_{\ell_0} \text{ subject to } \mathbf{y} = \Phi\mathbf{x}$$

Since the above problem is NP hard, in practice, one often works on the relaxed version

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} ||\mathbf{x}||_{\ell_1} \text{ subject to } \mathbf{y} = \Phi\mathbf{x}$$

which can be solved efficiently via linear programming. It turns out that if $\mathbf{x}$ is a k-sparse signal, then there exists a mapping Φ so that with an encoded vector $\mathbf{y}$ of dimension $O(k \log d)$, one can recover $\mathbf{x}$ from $\mathbf{y}$ through the above sparsity pursuit procedure [6]. Note that finding a good encoder Φ that works for any k-sparse signal $\mathbf{x}$ is not easy. Interestingly, it has been shown that a good encoder Φ should satisfy the so called restricted isometry property (RIP) that guarantees $\forall\mathbf{x}$, we have

$$(1-\epsilon)||\mathbf{x}||^2 \leq ||\Phi\mathbf{x}||^2 \leq (1+\epsilon)||\mathbf{x}||^2 \tag{2.4}$$

with striking similarity to (2.2). In fact, if we use random projection matrix Φ, then the probability that we fail the RIP property can be made exponentially small so it reveals that effective compressed sensing procedure exists [4].

Chapter 3

Analysis of Random Projection Algorithms

For dimensionality reduction methods that preserve the pairwise distance with low distortion, Johnson-Lindenstrauss lemma provides theoretical guideline on how well a metric can be embedded from a high dimensional space to a low dimensional one. We know that if the metric is Euclidean, then the embedding can be made with ϵ-distortion. For other metrics, it is usually worse: maybe with a constant or logarithmic distortion [11]. In fact, a space with ℓ_p-distance metric that satisfies the Johnson-Lindenstrauss lemma is very close being Euclidean: all of its n-dimensional subspaces are isomorphic to Hilbert space with distortion $2^{2^{O(\log n)}}$ [10]. Thus in the subsequent experimental study, we will focus on data points that have been be embedded onto an Euclidean space for performance evaluation with practical settings of d, n, and k.

3.1 Error Bound Analysis

From Lemma 2, it seems that if one needs to reduce the dimensionality of data from d to k, the lowest reduced dimension would be

$$k = c\frac{4+2\beta}{\epsilon^2/2 - \epsilon^3/3}\log n \tag{3.1}$$

where c is a positive constant not depending on d. However, it is possible that one may end up with values of n, ϵ and β so that $k > d$, i.e., the bound is meaningless. Thus it is important to know whether there are cases that we can not find any ϵ or β to reduce the original dimension through random project method.

Lemma 3: One can not reduce the dimension with arbitrary distortion level in the pairwise distance if and only if the number of data points n grows exponentially in d.

Proof: To fail the dimensionality reduction using Lemma 2, we need

$$d \geq c\frac{4+2\beta}{\epsilon^2/2 - \epsilon^3/3}\log n.$$

For $\epsilon \in (0, 1)$, we have

$$\epsilon^2/2 - \epsilon^3/3 < \frac{1}{6} \leq c\frac{4+2\beta}{d}\log n.$$

To ensure that random projection will fail $\forall \beta > 0$, we need

$$\log n > \frac{d}{24c}.$$

Clearly, n has to grow exponentially in d. On the other hand, when n grows exponentially in d, the original Johnson-Lindenstrauss lemma provides a trivial bound, thus the proof is complete.

3.2 Approximate PCA

Random projection can not only preserve the pairwise distance but also preserve the principal directions of the original data matrix with high probability. Let us assume that we want to transform d-dimensional data matrix A to k dimensional matrix B via random projection matrix R given by

$$B = \frac{1}{\sqrt{k}} AR$$

where each entry r_{ij} in R is drawn independently from standard Gaussian distribution, i.e., $r_{ij} \sim \mathcal{N}(0, 1)$. Assume that $d = \frac{c \log n}{\epsilon^2}$ where c is some appropriately chosen constant such that B preserves the pairwise distance within ϵ distortion. Now we compute the principal components of B via singular value decomposition such that

$$B = \sum_{i=1}^{k} \lambda_i \mathbf{a}_i \mathbf{b}_i^T \tag{3.2}$$

where $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_k$; $\mathbf{a}_i$ is the left singular vector corresponding to the singular value λ_i and $\mathbf{b}_i$ is the right singular vector, respectively. If we only use the best rank-L approximation B_L to B, then the PCA algorithm can be made very efficient when $k << d$. Let A_L be the best rank-L approximation to A and $\bar{A}_L$ be the approximate principal directions obtained via B_L, i.e.,

$$\bar{A}_L = A \sum_{i=1}^{L} \mathbf{b}_i \mathbf{b}_i^T \tag{3.3}$$

We would expect $\bar{A}_L$ to be a good approximation to A_L if the random projection satisfies the Johnson-Lindenstrauss lemma.

Theorem 1: With probability at least $1 - 4n^{-\frac{(\epsilon^2 - \epsilon^3)k}{4}}$, the approximation $\bar{A}_L$ satisfies

$$||A - \bar{A}_L||_F \leq ||A - A_L||_F + \epsilon ||A_L||_F \tag{3.4}$$

Proof: Suppose that the SVD yields

$$A = \sum_{i=1}^{d} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$$

$$B = \sum_{i=1}^{k} \lambda_i \mathbf{a}_i \mathbf{b}_i^T$$

and we have the approximate PCA given by

$$\bar{A}_L = A \sum_{i=1}^{L} \mathbf{b}_i \mathbf{b}_i^T.$$

We can see that

$$\begin{aligned}
||A - \bar{A}_L||_F &= \sum_i ||A\mathbf{b}_i - \bar{A}_L \mathbf{b}_i||^2 \\
&= \sum_i ||A\mathbf{b}_i - A(\sum_{j=1}^{L} \mathbf{b}_j \mathbf{b}_j^T)\mathbf{b}_i||^2 \\
&= ||A||_F - \sum_{i=1}^{k} ||A\mathbf{b}_i||^2 \\
&= ||A - A_L||_F + ||A_L||_F - \sum_{i=1}^{k} ||A\mathbf{b}_i||^2
\end{aligned}$$

We want to relate $\sum_{i=1}^{k} ||A\mathbf{b}_i||^2$ to the singular values of B. Note that with probability at least $1 - 4n^{-\frac{(\epsilon^2-\epsilon^3)k}{4}}$, we have [7]

$$\begin{aligned}
\sum_{i=1}^{L} \lambda_i^2 &= \sum_{i=1}^{L} ||B\mathbf{b}_i||^2 = \sum_{i=1}^{L} \frac{1}{k} ||AR\mathbf{b}_i||^2 \\
&\leq (1 + \epsilon/2) \sum_{i=1}^{L} ||A\mathbf{b}_i||^2
\end{aligned}$$

On the other hand, since $\mathbf{v}_i$ are the basis for A, with probability at least $1 - 4n^{-\frac{(\epsilon^2-\epsilon^3)k}{4}}$, we also have

$$\begin{aligned}
\sum_{i=1}^{L} \lambda_i^2 &\geq \sum_{i=1}^{L} ||B\mathbf{v}_i||^2 = \sum_{i=1}^{L} \frac{1}{k} ||AR\mathbf{v}_i||^2 \\
&\geq (1 - \epsilon/2) \sum_{i=1}^{L} ||A\mathbf{v}_i||^2 = (1 - \epsilon/2)||A_L||_F
\end{aligned}$$

Combining the above inequalities, we can lower bound $\sum_{i=1}^{k} ||A\mathbf{b}_i||^2$ by

$$\sum_{i=1}^{k} ||A\mathbf{b}_i||^2 \geq \frac{1 - \epsilon/2}{1 + \epsilon/2} ||A_L||_F \geq (1 - \epsilon)||A_L||_F$$

Thus one should be able to find the approximate principal directions of A from B_L with arbitrarily high probability by repeatedly constructing R until the ϵ-distortion is satisfied.

Chapter 4

Evaluation of Random Projection Algorithms

4.1 Motivation

We implemented two random projection methods for dimensionality reduction, namely, Achlioptas random projection method [1] and fast Johnson-Lindenstrauss transform (FJLT) [2]. For n data points of dimension d, we applied random project algorithms to reduce the dimension of each data point to k and computed the distances between data points before and after the transform. We are interested in knowing the acceptable k with desired distortion level ϵ for some n of practical relevance.

4.2 Performance Comparison

In the experimental study, we generated n data points $X_i \in R^d$ $(i = 1, 2, ..., n)$ with each element of X_i drawn independently from Gaussian distribution $\mathcal{N}(0, 1)$. We normalized each data point to have unit distance to the origin. We first implemented Achlioptas random projection algorithm [1] and computed the ratio of pairwise distances before and after the transform. From the histogram of the distance ratio, we chose the distortion tolerance ϵ such that 99% of the distance ratios fall within the interval $[1 - \epsilon, 1 + \epsilon]$. In addition, we need to control the sparsity q in implementing the fast Johnson-Lindenstrauss transform. We set $q = \frac{\log n}{d\epsilon}$. Note that the Johnson-Lindenstrauss lemma states that $k = O(\frac{\log n}{\epsilon^2})$ while we need to find out the scaling factor C such that $k = \frac{C \log n}{\epsilon^2}$ would be meaningful for n and ϵ in some reasonable ranges.

First, we set the number of data points $n = 50$. We applied Achlioptas random projection method to the input data with varying dimensions d from 1024 to 8192. 100 runs for each case were used to determine ϵ with different input dimension d and the results are summarized in Table 4.1. We can see that 99% of the ratios between the pairwise distances before and after the dimensionality reduction fall within the distortion level of 0.26 when $k = 100$.

We applied the same procedure for the cases of n=100 and n=200 with varying input dimension d from 1024 to 8192. The results are listed in Tables 4.2–4.3. Note that ϵ does not change significantly as d increases in all cases with fixed n and k. This confirms the theoretical statement of the Johnson-Lindenstrauss lemma. The scaling factor C varies slightly as n increases. Thus to reduce the high dimensional data to low dimension with

Table 4.1: Finding the appropriate scaling factor using Achlioptas random projection method (n=50)

d	k	ϵ	C
1024	100	0.24	1.5
2048	100	0.25	1.6
3072	100	0.25	1.6
4096	100	0.25	1.6
5120	100	0.25	1.6
6144	100	0.25	1.6
7168	100	0.26	1.7
8192	100	0.26	1.7

Table 4.2: Finding the appropriate scaling factor using Achlioptas random projection method (n=100)

d	k	ϵ	C
1024	100	0.25	1.4
2048	100	0.25	1.4
3072	100	0.25	1.4
4096	100	0.25	1.4
5120	100	0.25	1.4
6144	100	0.25	1.4
7168	100	0.25	1.4
8192	100	0.26	1.5

Table 4.3: Finding the appropriate scaling factor using Achlioptas random projection method (n=200)

d	k	ϵ	C
1024	100	0.26	1.3
2048	100	0.26	1.3
3072	100	0.26	1.3
4096	100	0.26	1.3
5120	100	0.26	1.3
6144	100	0.26	1.3
7168	100	0.26	1.3
8192	100	0.26	1.3

Table 4.4: Finding the acceptable k with various n and d for ϵ=0.4

n	d	k	ϵ	C
50	1024	29	0.4	1.2
50	2048	30	0.4	1.2
50	4096	31	0.4	1.3
100	1024	37	0.4	1.3
100	2048	38	0.4	1.3
100	4096	39	0.4	1.4
200	1024	43	0.4	1.3
200	2048	43	0.4	1.3
200	4096	44	0.4	1.3

Table 4.5: Finding the acceptable k with various n and d for ϵ=0.1

n	d	k	ϵ	C
50	1024	700	0.1	1.8
50	2048	700	0.1	1.8
50	4096	700	0.1	1.8
100	1024	780	0.1	1.7
100	2048	780	0.1	1.7
100	4096	780	0.1	1.7
200	1024	850	0.1	1.6
200	2048	850	0.1	1.6
200	4096	850	0.1	1.6

tolerable distortion around ϵ=0.26, the practical range for k is around 100 for n from 50 to 200. This seems to be still high for the method to be applicable to practical engineering applications.

We also applied fast Johnson-Lindenstrauss transform with multiple runs to find the lowest acceptable dimension k for different scenarios with different values of d, n at the desired distortion level ϵ=0.4. Note that ϵ was chosen based on Achlioptas random projection method with 99% pairwise distances being within the tolerable distortion range. The results are summarized in Table 4.4.

When ϵ=0.1, the results are summarized in Table 4.5 with the configurations similar to those in Table 4.4. We can see that the dimension does not reduce significantly when one imposes a more stringent distortion tolerance level.

Next, we consider the scenario where the number of data points n=50 and the dimension of the output k=100 after applying the Achlioptas random projection method and the fast Johnson-Lindenstrauss transform. We computed the pairwise distances before and after the dimensionality reduction transform and show the ratio of the pairwise distances using histogram in Fig. 4.1. Ideally, we expect the histogram to be centered around 1 with high concentration within $1 \pm \epsilon$. In order to determine the appropriate distortion tolerance level, we set the scaling factor C=1 and computed $\epsilon = \sqrt{\frac{\log n}{Ck}}$. We repeated the same procedure

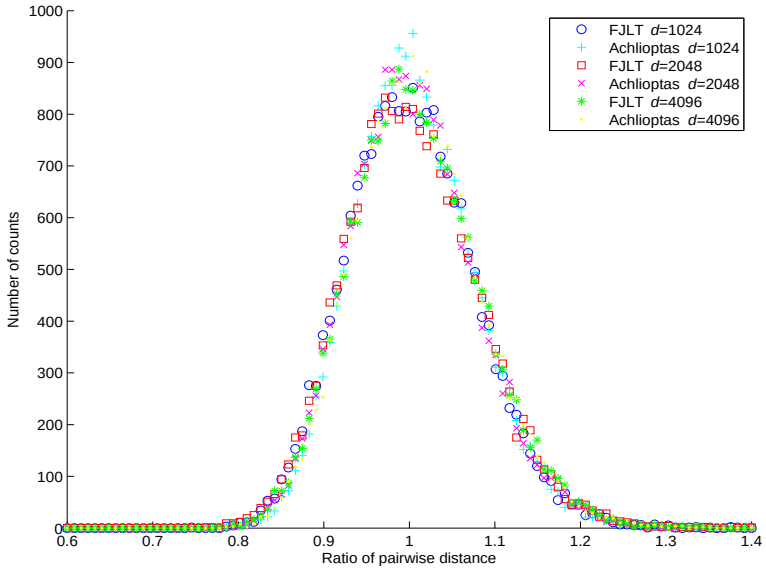

Figure 4.1: Histograms of the pairwise distance ratios, n=50, k=100, $\epsilon = 0.24$, Pr(pairwise distance in range)=0.99

for different scenarios by varying the dimension of each input data point d. From Fig. 4.1, we can see that the results using the fast Johnson-Lindenstrauss transform are similar to those using the Achlioptas random projection method. We found that with probability 0.99, we will preserve the pairwise distance within the distortion no greater than ϵ=0.24. This is in line with the results seen in Tables 4.1–4.3.

When n=100 and k=100, with varying dimension of each input data point d=1024, 2048 and 4096, the probability of preserving pairwise distance is 0.99 with distortion level ϵ=0.25. We can see from Fig. 4.2 that fast Johnson-Lindenstrauss transform yields similar histograms to those using the Achlioptas random projection method.

When n=200 and k=100, the performance comparison between fast Johnson-Lindenstrauss transform and Achlioptas random projection method is shown in Fig. 4.2. We found that with probability 0.99, we will preserve the pairwise distance within the distortion no greater than ϵ=0.26.

From the above comparison, we can see that FJLT has comparable performance to Achlioptas random projection method and the input dimension d does not affect the performance of both random project methods.

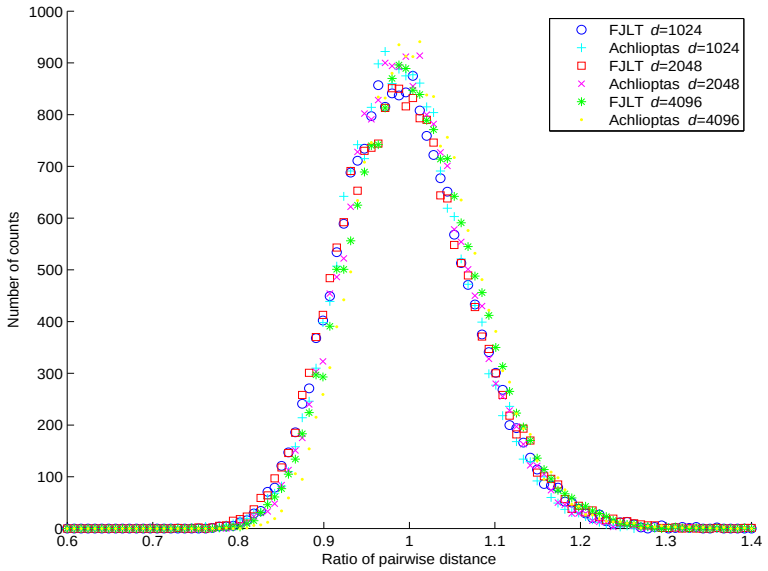

Figure 4.2: Histograms of the pairwise distance ratios, n=100, k=100, $\epsilon = 0.25$, Pr(pairwise distance in range)=0.99

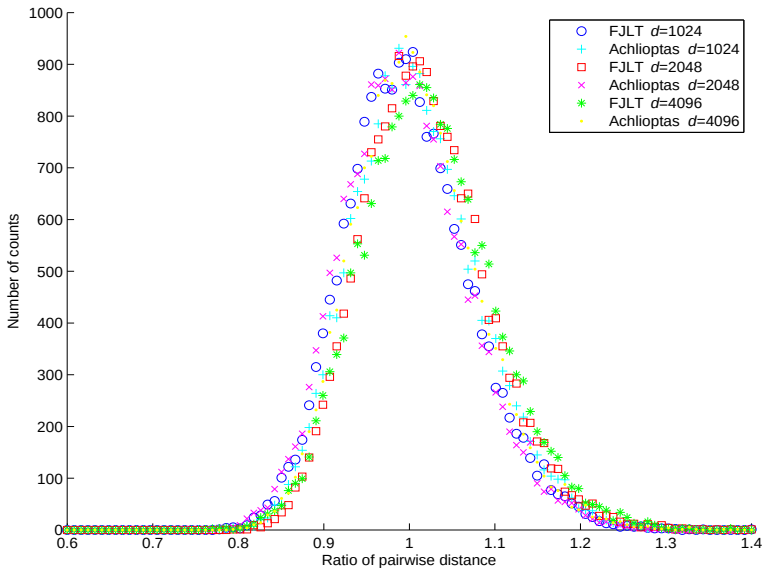

Figure 4.3: Histograms of the pairwise distance ratios, n=200, k=100, $\epsilon = 0.26$, Pr(pairwise distance in range)=0.99

Chapter 5

Conclusions and Future Work

5.1 Conclusions

Random projections are a powerful method of dimensionality reduction that provide us with both conceptual simplicity, and very strong error guarantees. The simplicity of projections allows them to be analyzed thoroughly, and this, combined with the error guarantees, makes them a very popular method of dimensionality reduction. In this thesis, we studied two random projection algorithms for dimensionality reduction that preserve the pairwise distances among the data points with small distortion. We found that the fast Johnson-Lindenstrauss transform has comparable performance to the Achlioptas random projection method with better computational efficiency. However, both methods have to sacrifice a small distortion for data points of moderate size.

5.2 Future work

There are many potential areas in which work can be carried out in the future. A brief list is compiled below.

- Lower bound analysis for the reduced dimension k, possibly using Ailon's method but under special cases, or perhaps a proof of a tight lower bound for specific input data
- An extended study of input distributions and their impact on the distortion of the projection, possibly trying in different ways of defining a distribution (using e.g., the moment generating function of the distribution instead of the mean/variance)
- Using the results derived on input sparsity to derive an analytical solution (or at least to identify some distinct cases) for the distortion as a function of the "true" vector
- Application of the random projection method to prognostic data for remaining useful life prediction of battery [12]

Bibliography

[1] D. Achlioptas, "Database friendly random projections", *Proc. of ACM Symoisium in Theory of Computing*, New York, NY, USA, pp. 274–281, 2001.

[2] N. Ailon, and B. Chazelle, "Approximate nearest neighbours and the fast Johnson Lindenstrauss transform", *Proc. of ACM Symposium on the Theory of Compututing*, Seattle, WA, USA, pp. 557–563, 2006.

[3] N. Ailon and E. Liberty, "Fast dimension reduction using Rademacher series on dual BCH codes", *Symposium on Discrete Algorithms*, San Francisco, CA, USA, 2008.

[4] R. Baraniuk, M. Davenport, R. Devore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices", *Contr. Approx.*, 2008.

[5] K. L. Clarkson, "Tighter bounds for random projection manifolds", *Proc. of 24th Annual Symposium on Computational Geometry*, New York, NY, USA, pp. 39–48, 2008.

[6] D. L. Donoho, "Compressed sensing", *IEEE Trans. on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[7] S. Dasgupta, and A. Gupta, "An elementary proof of the Johnson-Lindenstrauss lemma", Technical report 99-006, International Computer science Institute, Berkeley, 1999.

[8] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Springer, 2001.

[9] W. B. Johnson, and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space", *Contemporary Mathematics*, 26, pp. 189–206, 1984.

[10] W. B. Johnson, and A. Naor, "The Johnson-Lindenstrauss lemma almost characterizes Hilbert space, but not quite", *Proc. of ACM-SIAM Symposium on Discrete Algorithms*, Philadelphia, PA, USA, pp. 885–891, 2009.

[11] J. Matousek, "Bi-Lipschitz embeddings into low dimensional Euclidean spaces", *Comment. Math. Univ. Carolinae*, 31, pp. 589–600, 1990.

[12] M. C. Smart, B. V. Ratnakumar, K. B. Chin, L. D. Whitcanack, E. D. Davies, S. Surampudi, M. A. Manzo, P. J. Dalton, "Lithium-ion cell technology demonstration for future NASA applications", *Intersociety Energy Conversion Engineering Conference (IECEC)*, Washington, DC, USA, pp. 297–304, 2002.

Vita

Harika Rao Vamulapalli completed her Bachelor of Technology degree in Electronics and Communication Engineering from Jawaharlal Nehru Technological University, India in 2007. In 2008, she started her Masters in the Department of Electrical Engineering at the University of New Orleans. She is working as a Research Assistant with Dr. Huimin Chen. Her areas of interest include signal and data processing, machine learning with applications to fault prognosis.