Analysis of Variants of Johnson-Lindenstrauss lemma

Hrishikesh Vaidya Tapan Sahni
{cs13b1035,cs13b1030}@iith.ac.in

Abstract

Johnson-Lindenstrauss transform (JLT) projects n points in $\mathbf{R}^d$ to $\mathbf{R}^d$ where $k = O(\frac{\log n}{\epsilon^2})$ such that all pairwise distances are distorted by at most $1 \pm \epsilon$ epsilon multiplicative factor with probability greater than $1 - \frac{1}{2m}$. The lemma resides on dimensionality reduction which is very important tool in dealing with high dimensional data such as text. There has been little focus on the understanding of the embedding generated by different mapping matrices. In this report, we provide an empirical study on different variants on Johnson-Lindenstrauss lemma.

We provide a comprehensive analysis of the quality of the output as most of the methods used are randomised. We try to answer questions on the average running time of a specific variant. We show that "best in class" algorithm depends on the parameters chosen as well as the sparsity of the data. We also explore a new variant of JL lemma for circular matrices that minimises randomness and provide better running time.

1 Introduction

The Johnson–Lindenstrauss lemma(1) is the following. **Theorem 1.1** Let $\epsilon \in (0, \frac{1}{2})$ be a real number and let $P = \{p_1, p_2, \ldots, p_d\}$ be a set of n points in $\mathbf{R}^d$. Let k be an integer with $k \geq C\epsilon^{-2} \log n$, where C is a sufficiently large absolute constant. Then there exists a mapping $f : \mathbf{R}^n \to \mathbf{R}^k$ such that

$$(1 - \epsilon)\|p_i - p_j\| \leq \|f(p_i) - f(p_j)\| \leq (1 + \epsilon)\|p_i - p_j\|$$

for all $i, j = 1, 2, \ldots, n$ where $\|.\|$ denotes the Euclidean norm.

That is, every set of n points in the Euclidean space of any dimension d can be reduced to dimension $O(\epsilon^{-2} \log n)$ in in such a way that all distances between the points are preserved up to a multiplicative factor between $1 + \epsilon$ and $1 - \epsilon$ with a probability of at least $1 - \frac{1}{2n}$

We say that such a mapping is distance preserving. In most cases f is constructed by picking a random matrix $\Phi \in \mathbf{R}^{k \times d}$ is picked from a carefully chosen distribution and $f(x) = \Phi x$. Some of the variants will involve changing the probability distribution we sample from.

It is a very powerful method to tackle the “curse of dimensionality” that comes with high dimensional data. Indeed, there is growing interest in applying the JL lemma directly in application settings as diverse as natural language processing[4] which deals with high dimensional text data.
As the number of variants of JL lemma increases, there has been very little focus on studying these embedding on real world data. This is essential for determining the choice of the algorithm to use for specific data and what values of parameters fits the data the best. Our study here help to answer these questions.

1.1 Our work

- We demonstrated that for all algorithms considered, the norms are distributed as per the JL Lemma.

- We compared distortion graphs for different values of Matrix sparsity.
-

1.2 Report Outline

Our report is organised as follows. We will discuss the variants employed in Sections 2. We will briefly discus about implementation in Section 4 followed by results and analysis in Section 5.

2 Methods

We start by discussing about some of the major variants in JL Lemma.

2.1 Dense Projections

Indyk and Motwani(**?**) noted that the condition of orthogonality can be dropped, and in their proof, they choose the entries of A as independent random variables with the standard normal distribution $\mathcal{N}(0, 1)$ as in contrast to the original proof that required Φ to be orthonormal

Sampling all entries of Φ from a normal distribution yields a dense matrix. This can lead to slower running time for finding embeddings.

2.2 Sparse Projections

Achlioptas(**?**) improved on both these aspects. In his approach, Φ is constructed as a sparse matrix. Each element of Φ is set to 0 with probability $2/3$, and is set to 1 or -1 with equal probability $1/6$. While the resulting matrix is still not sparse, having $O(kd)$ nonzero entries, in practice it has few enough nonzero entries to take advantage of sparse matrix multiplication routines.

Matousek(**?**) proposed to use the parameter q to quantify sparsity. A distribution over matrices has sparsity q if each element of Φ is chosen to be 0 with probability $1 - \frac{1}{q}$ and $\frac{1}{2q}$ otherwise with equal probability. We will use q as a parameter in the results section to see the quality of projections as q increases.

2.3 Walsh Hadamard Transform

Ailon and Chazelle(**?**) came up with an extension of the idea of speeding up the evaluation of Φ by using a sparse random matrix. They defined the matrix Φ as the product of three matrices P, H, D.

- P is a matrix whose entries are independent random variables. They attain 0 with a probability of $1 - \frac{1}{q}$ and with probability $\frac{1}{q}$ entries are sampled from a gaussian distribution with mean 0 and variance $\frac{1}{q}$.
- H is an $n \times n$ Walsh matrix (assuming n to be a power of 2). To speed up the computations, we can evaluate $H \times x$ where x is the data matrix using a Fast Fourier Transform algorithm.
- D is a diagonal matrix with independent random ± 1 entries.

2.4 JL Lemma for Circulant Matrices

Hinrichs(**?**) proved a variant of a Johnson-Lindenstrauss lemma for matrices with circulant structure. This approach allows to minimise the randomness and implementation intricacies and provide better running time. The following notation is followed.
Let $a = (a_0, a_1, \ldots, a_{d-1})$ be independent identically distributed random variables. We denote by $M_{a,k}$ the partial circulant matrix

$$M_{a,k} = \begin{pmatrix} a_0 & a_1 & a_2 & \cdots & a_{d-1} \\ a_{d-1} & a_0 & a_1 & \cdots & a_{d-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{d-k+1} & a_{d-k+2} & a_{d-k+3} & \cdots & a_{d-k} \end{pmatrix}$$

Furthermore, if $\xi = (\xi_0, \ldots, \xi_{d-1})$ are independent Bernoulli variables, we set

$$D_\xi = \begin{pmatrix} \xi_0 & 0 & \ldots & 0 \\ 0 & \xi_1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \xi_{d-1} \end{pmatrix}$$

Then with probability at least $2/3$ the following holds

$$(1-\epsilon)\|x_i - x_j\| \leq \|f(x_i) - f(x_j)\| \leq (1+\epsilon)\|x_i - x_j\|$$

2.5 Dimension Reduction for L1 norm

We try to answer the following question. Is there an analogue of JL lemma for L-1 norm? Moses(**?**) show that for the l-1 norm, one can not hope to use linear embeddings as a dimensionality reduction tool for general point sets, even if we select linear embedding as the function for the given set. There is a distribution what has the stability for the l-1 norm. The Cauchy distribution(**?**) is defined as

$$p(s) = \frac{1}{\pi(s^2+1)}$$

One interesting property for the Cauchy distribution that it stable for l1-norm is

$$\sum c_i z_i \sim c_i \dot{\|} z\|_1$$

We have also shown it in our analysis that Cauchy distribution is able to preserve l1 norm.

3 Experiments

The experiments were carried out by employing the above mentioned methods to generate the transform matrix. The analysis sections consists comparison based on quality of projections and matrix sparsity.

3.1 Setup

For testing, the value of k was set to $\frac{2\ln n}{\epsilon^2}$. We took $\epsilon = 0.25$ for all set of experiments. For comparing quality of projections, number of data points were set to 1000.

3.2 Data Generation

The experiments were peformed on two types of data namely dense and sparse. For dense data, each element was uniformly sampled at random from $[-1, 1]^d$. Sparse data was generated by selecting 10 dimensions randomly and then each dimension was populated as above.

4 Results

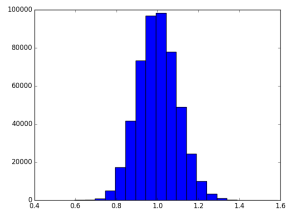
This section contains the analysis performed by performing dimensionality reduction using the above algorithms. Specifically the analysis comprises of evaluation of quality of projections, distortions with increasing matrix sparsity and running time comparison.

4.1 Quality of projections

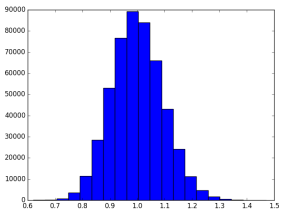
The quality of projections is evaluated based on the distortion incurred which is basically the $l2$ norm of each pair of points.

4.2 Distortions with increasing epsilon

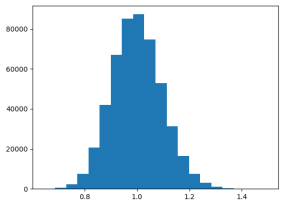
Here we plot the distortion error for dense and sparse using dense transform algorithm for different values of ϵ



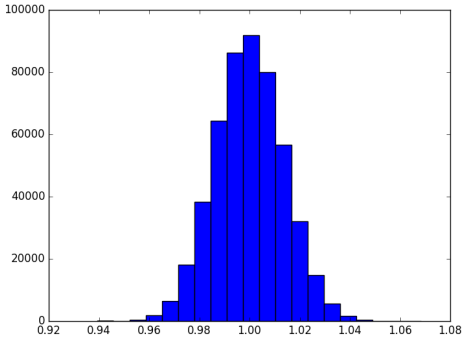
(a) Dense



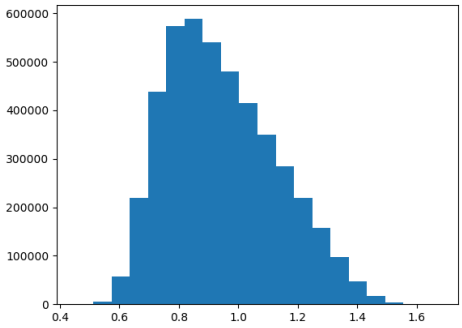
(b) Sparse



(c) Walsh Hadamard Transform



(a) Circulant



(b) Reduction with ℓ_1 Norm

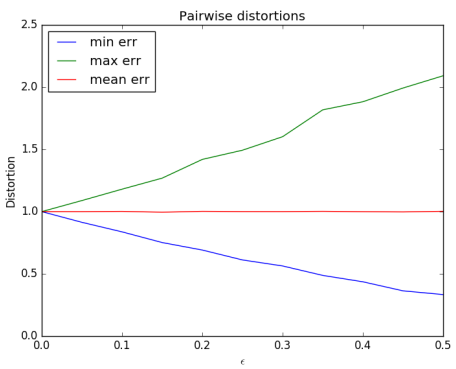
4.3 Distortions with increasing matrix sparsity

In this subsection, we discuss about the stability of distortions with increasing sparsity. For Sparse JL, as the value of q increases the projections become unstable. The projections are even worse in case of sparse data. This is due to the construction of sparse data where only few dimensions contain non-zero values. Multiplication of sparse matrix with a sparse vector may give a vector whose $l2$-norm tends to zero.

4.4 Comparison of running time

The running time was compared for three different algorithms namely dense, sparse and circulant. Due to higher computational complexity of Hadamard transform, we were unable to obtain the running time for the same. The running times were obtained for different values of n where n is the number of data points.

Figure 3: Distortion plots for Dense and Sparse Matrices for different values of ϵ



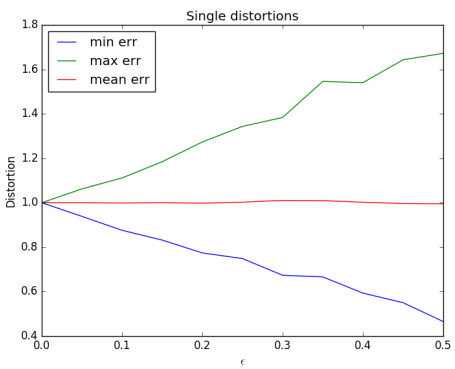
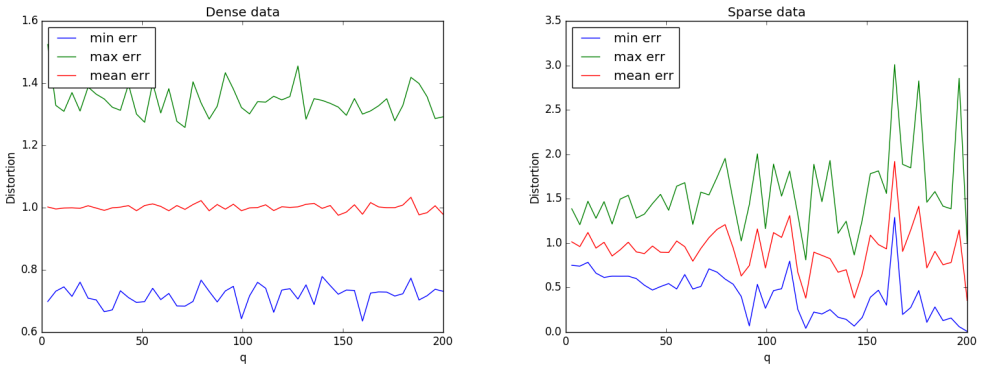


Figure 4: Distortion plot for different values of q



From the plot, it can be observed that as the value of n increases, sparse and circulant transform tend to perform better than dense transform.

References

[1] S. Venkatasubramanian, & Q. Wang. (2011) The Johnson-Lindenstrauss transform: An empirical study , *In ALENEX, 2011.*

[2] Matousek J (2008) On variants of the Johnson–Lindenstrauss lemma., *Random Struct. Algorithms 33(2)*, pp. 142-156.

[3] W. Johnson & Mozer, M.C. (1995) Extensions of Lipschitz mappings into a Hilbert space. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Conference in modern analysis and probability*, pp. 609-616.

[4] S. Ben-David & J. Blitzer (2006) Template-based algorithms for connectionist rule extraction. *Analysis of representations for domain adaptation*, pp. 137–144. NIPS

[5] P. Indyk & R. Motwani. (1998) Approximate nearest neighbors: Towards removing the curse of dimensionality. *Proc 30th ACM Symp Theory of Computing*, pp. 604-611

[6] D. Achlioptas. (2003) Database-friendly random projections: Johnson-lindenstrauss with binary coins. *J. Comput. Syst. Sci.*, pp. 681-697.

[7] J. Matousek. (2008) On variants of the johnson–lindenstrauss lemma. Random Struct. Algorithms, pp. 142-155.

[8] N. Ailon & B. Chazelle. (2007) Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform,, *Proc 38th ACM Symp Theory of Computing, 2006*, pp. 557–563

[9] Ping Li & Trevor J. Hastie. (2007) Nonlinear Estimators and Tail Bounds for Dimension Reduction in $l1$ Using Cauchy Random Projections, *Journal of Machine Learning Research 8*, pp. 2495-2513

[10] Moses Charikar & Amit Sahai (2002) Dimension Reduction in the ℓ_1 Norm, *Proceedings of the 43rd Symposium on Foundations of Computer Science*, pp. 551-560.

[11] A. Hinrichs & J. Vybıral. Johnson-Lindenstrauss lemma for circulant matrices.