Extremely Sparse Johnson-Lindenstrauss Transform: From Theory to Algorithm

Rong Yin*†, Yong Liu‡, Weiping Wang* and Dan Meng*
*Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100195, China
†School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China
‡Beijing Key Laboratory of Big Data Management and Analysis Methods,
Gaoling School of Artificial Intelligence, Renmin University of China, Beijing 100872, China
Emails: yinrong@iie.ac.cn, liuyonggsai@ruc.edu.cn, wangweiping@iie.ac.cn, mengdan@iie.ac.cn

***Abstract*—Dimension reduction is a fundamental data mining task. However, it has limited applicability in high-dimensional scenarios because of stringent computational requirements. To address these issues, we propose ESE, an extremely sparse Johnson-Lindenstrauss transform, which takes a substantial step in dimension reduction. The projection matrix of ESE is an extremely sparse matrix, which has only k nonzero elements by employing the hash functions, where k is the embedded dimension. Theoretical analysis shows that ESE has a smaller time complexity than the existing projection algorithms and keeps the best accuracy $(1+\varepsilon)$ for the general case, where $0 < \varepsilon \ll 1$. In particular, the optimal statistical accuracy is achieved requiring $\log(n)\log(d)/\varepsilon$ embedded dimension, where n is the number of data, d is the dimension of data. The extensive experiments verify that ESE has a significant advantage in time with satisfactory accuracy, compared to the state-of-the-art dimension reduction algorithms.**

***Index Terms*—dimension reduction; Johnson-Lindenstrauss; embeddings; extremely sparse;**

I. Introduction

For a high-dimensional dataset, directly processing datasets requires high time and storage space. Overcoming these limitations has motivated a variety of dimension reduction approaches to improve time complexity [1]–[7]. In data mining community, dimension reduction, mainly including feature extraction and feature selection, has become an important research and achieved remarkable achievement [8]–[15].

The singular value decomposition (SVD) feature selection [16] and approximate SVD [15] have a solid theoretical guarantee. However, their expensive calculations are not to be underestimated. Subsequently, a series of random projections (RP) algorithms were devised [15], [17], [18]. The elements of the projection matrices are mainly scaled based on $\{-1, 0, -1\}$ or $\{-1, 1\}$ at a certain probability. However, when data dimensions become more and more larger, the time consumed by the matrix multiplication in RP will still be prohibitive. Subsequently, Liu et al. [19] proposes a sparse embeddings algorithm (SE) which builds a sparse embedded matrix. SE has a faster computing speed $\mathcal{O}(\varepsilon^{-2}nd)$ with the embedded dimension $\mathcal{O}\left(K/\varepsilon^2\right)$ than the previous dimension reduction algorithms with satisfactory performance, where n and d are the number and dimension of data, and $0 < \varepsilon \ll 1$.

Yong Liu: Corresponding author

Unfortunately, for large d, the cost of matrix multiplication is still relatively large, and its accuracy $(1+\varepsilon)$ is only guaranteed for K-means. Makarychev et al. [20] also designed a random projections algorithm based on orthogonal and Gaussian projections, whose time complexity is $\mathcal{O}(\varepsilon^{-2}nd\log(K/\varepsilon))$ with the accuracy $(1+\varepsilon)$. The above is just some related research. For a broad overview, srefer to [21] and references therein.

In order to overcome the shortcomings, we propose an extremely sparse Johnson-Lindenstrauss transform ESE, a novel algorithm that, to the best of our knowledge, has the best known theoretical guarantees. ESE is based on hash functions to achieve sparsity. In matrix-matrix product, one only needs to multiply and store the nonzero elements. Therefore, the sparse matrix in ESE can greatly speed up the matrices operations. In detail, with time complexity $\mathcal{O}\left(\varepsilon^{-1}n\log(n)\log(d)\right)$ for fast matrix multiplication, ESE can keep the best accuracy $(1+\varepsilon)$ for general case. Compared to SE [19], ESE reduces the running time by a factor of $\mathcal{O}(d/(\varepsilon\log(d)\log(n)))$ and the embedded dimension by a factor of $\mathcal{O}(K/(\varepsilon\log(d)\log(n)))$. ESE in matrix multiplication is much faster than the state-of-the-art algorithms. Theoretical reasoning and experimental results demonstrate that ESE has a significant advantage over time complexity and keeps the best approximate accuracy.

Section 2 and 3 are the related work and proposed algorithm. Following sections are experiment, proof, conclusions and acknowledgment.

II. Related Work

Embeddings have become an important tool for dimension reduction. A real gem in this area has been the result of Johnson and Lindenstrauss [26]. The key of JL-projections is the design of projection matrix. At first, the projection matrices are dense or comparatively dense [17], [27]. For example, the entries of the projection matrix obey Gauss distribution or Bernoulli distribution. Due to the high density and projection dimension of the projection matrices, the time complexity of matrix operation is still high. For further reducing the computational requirements, the projection matrix becomes more and more sparse [15], [19], [24], [25], which can speed up the matrices operations to a certain extent. However, when the data dimension is high, the time cost is still costly.

2374-8486/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDM50108.2020.00180

TABLE I

COMPARISON OF THE CLASSICAL ALGORITHMS OF DIMENSION REDUCTION. THE SECOND, THIRD AND FOURTH COLUMNS CORRESPOND TO THE NUMBER OF SELECTED FEATURES, TIME COMPLEXITY AND APPROXIMATION ACCURACY. "(K-MEANS)" DENOTES THAT THE APPROXIMATION ACCURACY IS OBTAINED BY APPLYING THE DIMENSION REDUCTION ALGORITHM TO K-MEANS LEARNER, RATHER THAN ITS GENERAL ACCURACY. N/A, ε AND δ DENOTE "NOT AVAILABLE", THE GAP TO OPTIMALITY AND THE CONFIDENCE LEVEL, RESPECTIVELY.

Algorithm	Dimensions	Time	Accuracy
RP (FOLKLORE)	$\mathcal{O}(\log(n)/\varepsilon^2)$	$\mathcal{O}(\varepsilon^{-2}nd\log(n)/\log(d))$	$1+\varepsilon$
SVD [22]	K	$\mathcal{O}(nd\min\{n,d\})$	2 (K-clustering)
SVD [23]	$\mathcal{O}(K/\varepsilon^2)$	$\mathcal{O}(nd\min\{n,d\})$	$1+\varepsilon$(K-clustering)
RP [24]	$\mathcal{O}(\log(n)/\varepsilon^2)$	$\mathcal{O}(\varepsilon^{-3}n\log^2(n))$	$1+\varepsilon$
RP [15]	$\mathcal{O}(K/\varepsilon^2)$	$\mathcal{O}(\varepsilon^{-2}ndK/\log(d))$	$2+\varepsilon$(K-means)
RP [25]	$\mathcal{O}(\log(n)/n)$	$\mathcal{O}(nd\log(d))$	N/A
SE [19]	$\mathcal{O}\left(K/\varepsilon^2\right)$	$\mathcal{O}(\varepsilon^{-2}nd)$	$1+\varepsilon$(K-means)
RP [20]	$\mathcal{O}\left(\log(K/\varepsilon)/\varepsilon^2\right)$	$\mathcal{O}(\varepsilon^{-2}nd\log(K/\varepsilon))$	$1+\varepsilon$(K-means)
ESE (This Paper)	$\mathcal{O}\left(\log(n)\log(d)/\varepsilon\right)$	$\mathcal{O}\left(\varepsilon^{-1}n\log(n)\log(d)\right)$	$1+\varepsilon$

Inspired by those, we construct an extremely sparse Johnson-Lindenstrauss transform (called ESE), which skillfully uses hash functions to construct the projection matrix to achieve sparsity. By theoretical analysis, compared to the state-of-the-art dimension reduction algorithms, ESE is not only more sparse in the projection matrix but also has a smaller projection dimension $\mathcal{O}\left(\log(n)\log(d)/\varepsilon\right)$ with the optimal statistical accuracy $(1+\varepsilon)$. To the best of our knowledge, this is the first time that all these achievements have been achieved. Experimental results and theoretical reasoning verify that ESE has much less time complexity with the satisfactory accuracy. Table I shows the detail of the related algorithms.

III. EXTREMELY SPARSE JOHNSON-LINDENSTRAUSS TRANSFORM

In this section, we characterize the properties of ESE showing it achieves the optimal statistical accuracy, with dramatically reduced computations. This main result is given in Theorem 1. The complexity analysis also follows.

A. Proposed Algorithm

ESE can be described in general using hash-based projection matrix. Let h be hash function from $\{1,\ldots,k\}$ to $\{1,\ldots,d\}$, $h(j)=i$ for $i\in\{1,\ldots,d\}$ with probability of $1/d$, and σ be 2-wise independent hash function from $\{1\ldots,d\}$ to $\{-1,+1\}$, $\sigma(i)=j$ for $j\in\{-1,+1\}$ with probability of $1/2$. Denote the projection matrix by $\mathbf{R}\in\mathbb{R}^{d\times k}$, where

$$\mathbf{R}_{ij}=\begin{cases}\sigma(i), & \text{for } i=h(j),\\ 0, & \text{for } i\neq h(j).\end{cases} \tag{1}$$

The projected dataset can be written as: $\hat{\mathbf{X}}=\sqrt{\frac{d}{k}}\mathbf{X}\mathbf{R}$, where $\mathbf{X}\in\mathbb{R}^{n\times d}$ is the dataset. $\mathbf{R}$ has only k nonzero elements, which is a sparse matrix. In the matrix-matrix product, we only need to multiply and store the nonzero elements. The detail of ESE is given in Algorithm 1.

Algorithm 1 Extremely Sparse Johnson-Lindenstrauss Transform (ESE)

Input: Dataset $\mathbf{X}\in\mathbb{R}^{n\times d}$.
Output: Embedding dataset $\hat{\mathbf{X}}\in\mathbb{R}^{n\times k}$.
1: Set $k=\mathcal{O}\left((2\log(n)-\log(\delta))\log(d)/\varepsilon\right)$.
2: Build hash function h: $\{1,\ldots,k\}\mapsto\{1,\ldots,d\}$.
3: Build hash function σ: $\{1\ldots,d\}\mapsto\{-1,+1\}$.
4: Build a matrix $\mathbf{R}\in\mathbb{R}^{d\times k}$, with $\mathbf{R}_{i,j}=\sigma(i)$ for $i=h(j)$ and $\mathbf{R}_{i,j}=0$ for $i\neq h(j)$.
5: Compute $\hat{\mathbf{X}}=\sqrt{\frac{d}{k}}\mathbf{X}\mathbf{R}$.

B. Complexity Analysis

In ESE, the projection matrix $\mathbf{R}$ based on hash functions is extremely sparse. The number of nonzero elements in the matrix $\mathbf{R}$ is only k. Therefore, the time complexity of the proposed algorithm is $\mathcal{O}(nk)$, while for the traditional projection algorithm with the dense matrix is $\mathcal{O}(ndk)$. Thus, the proposed ESE is much faster than the traditional projection when d is large.

C. Theoretical Analysis

In this part, we introduce Theorem 1 which is the main result and guarantees the effectiveness of ESE.

Theorem 1: Let $\mathbf{P}$ be an arbitrary set of n points in $\mathbb{R}^d$, represented as an $n\times d$ matrix $\mathbf{X}$. Given $\varepsilon,\delta\in(0,1)$, let

$$k_0=\mathcal{O}\left(\frac{(2\log(n)-\log(\delta))\log(d)}{\varepsilon}\right). \tag{2}$$

For integer $k\geq k_0$, let $\mathbf{R}$ be a $d\times k$ random matrix, which is constructed as in algorithm 1. Let

$$\hat{\mathbf{X}}=\sqrt{d/k}\mathbf{X}\mathbf{R} \tag{3}$$

and let $f:\mathbb{R}^d\to\mathbb{R}^k$ map the ith row of $\mathbf{X}$ to the ith row of $\hat{\mathbf{X}}$. With probability at least $1-\delta$, for all $\mathbf{u},\mathbf{v}\in\mathbf{P}$

$$(1-\varepsilon)\|\mathbf{u}-\mathbf{v}\|^2\leq\|f(\mathbf{u})-f(\mathbf{v})\|^2\leq(1+\varepsilon)\|\mathbf{u}-\mathbf{v}\|^2.$$

TABLE II
DATASETS USED IN THIS PAPER.

Data set	Instance	Feature	Class
ORL	400	4,096	40
gisette	6,000	5,000	2
TDT2	9,394	36,771	30
news20	15,935	62,061	20
Rcv1-binary	20,242	47,236	2
smallNORB	24,300	18,432	5
cifar10	50,000	3,072	10
SVHN	73,257	3,072	10

From a theoretical perspective, we know that this JL-projections allows one to construct a sparse projection matrix $\mathbf{R}$ using far fewer random nonzero entries than all previous algorithms. By the sparse projection matrix, we can project data into k-dimensional subspace and maintain the Euclidean distance in high accuracy. If the projection dimension $k \geq \mathcal{O}\big(\log(n)\log(d)/\varepsilon\big)$, the proposed algorithm preserves the optimal statistical accuracy $(1+\varepsilon)$, which demonstrates that the proposed algorithm is effective.

IV. EXPERIMENT

K-means is a widely recognized algorithm and reflects the distance relationship between data. ESE reflects that the distance relationship between the projected data is still approximately the same as the original one. Therefore, experiments analyze the performance of ESE based on K-means, and compare it with state-of-the-art algorithms: 1) SVD: A classic dimension reduction method; 2) RP: Choosing the representative one [15]; 3) SE[1]: the state-of-the-art random projections method [19]; 4) K-means [2], on the machine with 32 cores (2.40GHz) and 64 GB of RAM.

The experiments are conducted on 8 real-world datasets. See Table II for details. ORL and TDT2 are from website[3], and the rest are from website[4]. On each dataset, 70% is used for training and 30% for testing, and we normalize them. The number of clusters and classes is the same.

A. Evaluation Metrics

In order to avoid contingency, every kind of experiment is repeated 30 times. Under different k, we compute the projection time (in seconds), and then logarize it to get Fig. 2. After dimension reduction, run all algorithms on a standard K-means to get clustering accuracy (Fig. 1).

The accuracy is defined as $Accuracy = \frac{\sum_{i=1}^{n}\mu(\hat{y},map(y))}{n}$, where $\hat{y}$ is the ground truth label of the ith datum and y corresponds to the derived label. $\mu(p,q)$ is the delta function where $\mu(p,q)=1$ if $p=q$ and $\mu(p,q)=0$ otherwise. $map(\cdot)$ is the best mapping function that matches the true labels and the derived ones. The greater the accuracy, the better.

B. Experimental Results

1) Fig. 1 shows that ESE is at the same level of accuracy as the best. The accuracy of SE and RP fluctuate on different datasets. SVD maintains high accuracy on most datasets, which is almost the same as ESE. The higher the projection dimension is, the higher the clustering accuracy is. This verifies our theoretical results.

2) Fig. 2 shows ESE has a significant advantage over projection time and even achieves several hundred orders of magnitude faster than RP and others. The larger the dimension of the dataset, the more obvious the time advantage of ESE is. For example, on the higher dimension TDT2, news20 and Rcv1-binary datasets, ESE reduces the time cost by a factor of 400 compared to SE and $22,000$ compared to RP. With the increase of the projection dimension k, the projection time increases in every algorithm. These are consistent with the empirical results and our theoretical analysis.

V. PROOF

Let $\mathbf{x}\cdot\mathbf{y}$ denote the inner product of vectors $\mathbf{x},\mathbf{y}$. To simplify notation we will work with a matrix $\mathbf{R}$ scaled by d. As a result, to get $\hat{\mathbf{X}}$ we need to scale $\mathbf{XR}$ by $1/d$. Different columns of $\mathbf{R}$ are independent of each other. After scaling, the entry of $\mathbf{R}$ is $\mathbf{R}_{ij}$, where $\mathbf{R}_{ij} = dr_{ij}$. $\mathbf{c}_j$ denotes the jth column of $\mathbf{R}$, $\hat{\mathbf{x}}_i$ and $\mathbf{x}_i$ denote the ith row of $\hat{\mathbf{X}}$ and $\mathbf{X}$, namely the ith datum. Therefore, $\hat{\mathbf{x}}_i = \frac{1}{\sqrt{dk}}(\mathbf{x}_i\cdot\mathbf{c}_1,\ldots,\mathbf{x}_i\cdot\mathbf{c}_d)$. We replace $\mathbf{x}_i$ with $\mathbf{x}$. For $j=1,\ldots,k$, let $Q_j = Q_j(\mathbf{x}) = \mathbf{x}\cdot\mathbf{c}_j$.

Lemma 1: For all $h\in[0,1/(2d))$, all $d\geq 1$ and all unit vectors $\mathbf{x}$,

$$\mathbb{E}(\exp(hQ_1(\mathbf{x})^2)) \leq \frac{1}{\sqrt{1-2hd}}, \tag{4}$$

and

$$\mathbb{E}(Q_1(\mathbf{x})^4) \leq 3d^2. \tag{5}$$

Proof According to Lemma 2, we know $\mathbb{E}(Q(\mathbf{x})^4) \leq \mathbb{E}(\mathbf{T}^4)$, while $\mathbb{E}(\mathbf{T}^4) = \int_{-\infty}^{+\infty}\frac{1}{\sqrt{2\pi}}\exp(-\lambda^2/2)\left(\lambda^4 d^2\right)d\lambda = 3d^2$. The following will prove (4).

For any real-valued random variable U and for all h such that $\mathbb{E}(\exp(hU^2))$ is bounded. According to the Monotone Convergence Theorem (MCT), we get the formula $\mathbb{E}(\exp(hU^2)) = \mathbb{E}\left(\sum_{k=0}^{\infty}\frac{(hU^2)^k}{k!}\right) = \sum_{k=0}^{\infty}\frac{h^k}{k!}\mathbb{E}(U^{2k})$. Here we obtain:

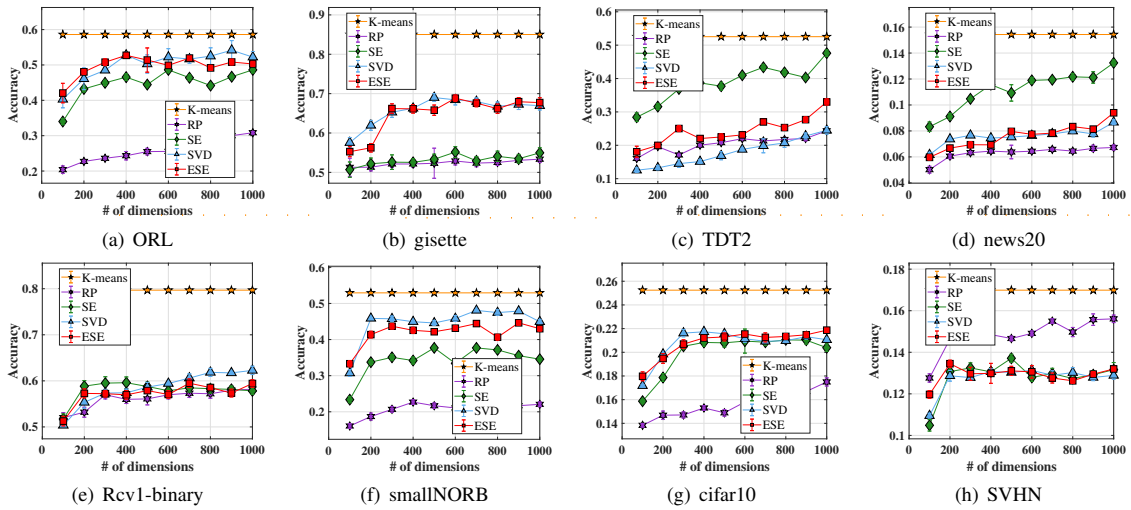
$$\begin{aligned}
&\mathbb{E}(\exp(hT^2))\\
&= \int_{-\infty}^{+\infty}\frac{1}{\sqrt{2\pi}}\exp(-\lambda^2/2)\exp(h\lambda^2 d)d\lambda\\
&= \frac{1}{\sqrt{1-2hd}} = \sum_{k=0}^{\infty}\frac{(h^k)}{k!}\mathbb{E}(T^{2k})\\
&\geq \sum_{k=0}^{\infty}\frac{(h^k)}{k!}\mathbb{E}(Q(\mathbf{x})^{2k}) = \mathbb{E}(\exp(hQ(\mathbf{x})^2)).
\end{aligned} \tag{6}$$

[1]The code is from website https://sites.google.com/site/weiweiliuhomepage/ with default parameters

[2]The code is from website www.cad.zju.edu.cn/home/dengcai/Data/data.html with default parameters

[3]http://www.cad.zju.edu.cn/home/dengcai/Data/data.html

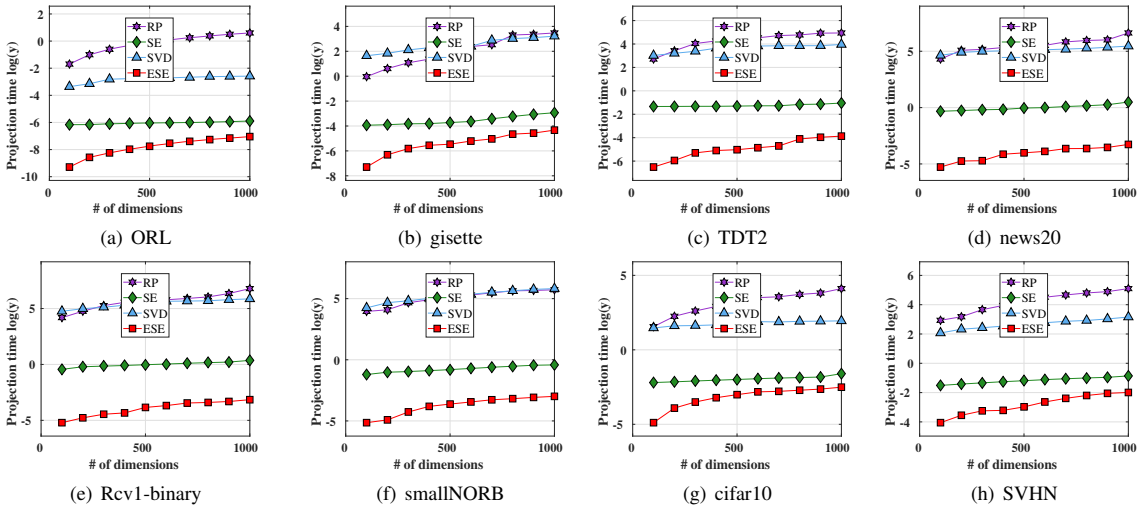
[4]http://www.csie.ntu.edu.tw/∼cjlin/libsvm.



(a) ORL (b) gisette (c) TDT2 (d) news20

(e) Rcv1-binary (f) smallNORB (g) cifar10 (h) SVHN

Fig. 1. Clustering accuracy and different k of various algorithms on ORL, gisette, TDT2, news20, Rcv1-binary, smallNORB, cifar10 and SVHN datasets.



(a) ORL (b) gisette (c) TDT2 (d) news20

(e) Rcv1-binary (f) smallNORB (g) cifar10 (h) SVHN

Fig. 2. Projection time and different k of various algorithms on ORL, gisette, TDT2, news20, Rcv1-binary, smallNORB, cifar10 and SVHN datasets.

For converge, we take $h \in [0, 1/(2d))$ and apply the MCT in (6). Therefore, we have $\mathbb{E}(\exp(hQ(\mathbf{x})^2)) \leq \frac{1}{\sqrt{1-2hd}}$, for $h \in [0, 1/(2d))$. □

Lemma 2: Let $\mathbf{T} \sim \mathcal{N}(0, d)$. For every unit vector $\mathbf{x} \in \mathbb{R}^d$, all $d \geq 1$ and all $k \in \mathbb{N}_+^0$, we have

$$\mathbb{E}(Q(\mathbf{x})^{2k}) \leq \mathbb{E}(\mathbf{T}^{2k}). \tag{7}$$

Proof Our strategy for giving bounds on the moments of $Q(\mathbf{x})$ will be to determine a "worst-case" unit vector $\mathbf{w}$ and bound the moments of $Q(\mathbf{w})$. Let $\mathbf{w} = \frac{1}{\sqrt{d}}(1, \ldots, 1)$. For any vector $\mathbf{x}$, $Q(\mathbf{x}) = Q_1(\mathbf{x}) = \mathbf{x} \cdot \mathbf{c}_1$, where $\mathbf{c}_1 = d(r_{11}, \ldots, r_{d1})$. If $\mathbf{x} = (x_1, \ldots, x_d)$ is such that $x_i^2 = x_j^2$ for all i, j, then by symmetry, $Q(\mathbf{x})$ and $Q(\mathbf{w})$ are identically distributed and this Lemma holds trivially. Otherwise, we can assume without loss of generality, that $x_1^2 \neq x_2^2$ and consider the "more balanced" unit vector $\theta = (c, c, x_3, \ldots, x_d)$, where $c = \sqrt{(x_1^2 + x_2^2)/2}$.

We express $\mathbb{E}(Q(\mathbf{x})^{2k})$ as a sum of averages over r_{11}, r_{21} and apply (8) in Lemma 3 to get that each term (average) in the sum. More precisely,

$$\begin{aligned}
&\mathbb{E}(Q(\mathbf{x})^{2k}) \\
&= d^{2k} \sum_M \mathbb{E}\left((M + x_1 r_{11} + x_2 r_{21})^{2k}\right) \cdot \mathbb{P}\left[\sum_{i=3}^{d} x_i r_{i1} = M \cdot d\right] \\
&\leq d^{2k} \sum_M \mathbb{E}\left((M + c r_{11} + c r_{21})^{2k}\right) \cdot \mathbb{P}\left[\sum_{i=3}^{d} x_i r_{i1} = M \cdot d\right] \\
&= \mathbb{E}(Q(\theta)^{2k}).
\end{aligned}$$

Applying this argument repeatedly yields the lemma, as θ eventually becomes $\mathbf{w}$, we obtain $\mathbb{E}(Q(\mathbf{x})^{2k}) \leq \mathbb{E}(Q(\mathbf{w})^{2k})$.

The following we will prove $\mathbb{E}(Q(\mathbf{w})^{2k}) \leq \mathbb{E}(\mathbf{T}^{2k})$.

$\mathbf{T} \sim \mathcal{N}(0,d)$, $\mathbf{T} = \sqrt{d}\hat{\mathbf{T}}$ and $\hat{\mathbf{T}} \sim \mathcal{N}(0,1)$. $Q(\mathbf{w}) = \mathbf{w}\cdot\mathbf{c}_1$. We have $Y \in \{-1,+1\}$ represents the nonzero elements in the first column of $\mathbf{R}$. Therefore, we get $Q(\mathbf{w}) = \sqrt{d}Y$.

For every $k \in \mathbb{N}_+$, we observe $\mathbb{E}(Q(\mathbf{w})^{2k}) = \Big(\sqrt{d}\Big)^{2k}\mathbb{E}(Y^{2k})$, and $\mathbb{E}(\mathbf{T}^{2k}) = \Big(\sqrt{d}\Big)^{2k}\mathbb{E}(\hat{\mathbf{T}}^{2k})$. Because $Y \in \{-1,+1\}$, we have $\mathbb{E}(Y^{2k}) = 1$. According to the well-known fact, we have the $(2k)$-th moment of $\mathcal{N}(0,1)$ is $(2k-1)!! = \frac{(2k)!}{k!2^k} \geq 1$. So, we get this lemma. □

Lemma 3: Let r_1, r_2 be any two numbers in $\{r_{ij}\}$. For any $a, b \in \mathbb{R}$, let $c = \sqrt{(a^2+b^2)/2}$. Then $\forall M \in \mathbb{R}$ and $k \in \mathbb{N}_+^0$,

$$\mathbb{E}\left((M + ar_1 + br_2)^{2k}\right) \leq \mathbb{E}\left((M + cr_1 + cr_2)^{2k}\right). \quad (8)$$

Proof We have the following representation, $W = \mathbb{E}((M + cr_1 + cr_2)^{2k}) - \mathbb{E}((M + ar_1 + br_2)^{2k}) = W_1 + W_2$. Because $\mathbf{R}_{ij} = dr_{ij}$ and r_1, r_2 belong to $\{r_{ij}\}$, we have $r_i \in \{0,+1,-1\}$. According to analysis we know $\mathbb{P}(r_i = 0) > \mathbb{P}(r_i = \pm 1)$, and $\mathbb{P}(r_i = +1) = \mathbb{P}(r_i = -1)$.

If $a^2 = b^2$, then $a = c$ and the lemma holds with equality. If $a^2 \neq b^2$, we have the following according to W_1 and W_2.

(1) W_1 is the value under the condition of $|r_1| = |r_2|$.

Remembering $\mathbb{P}(r_1 = r_2)$ as $\mathbb{P}_{11}$, we get $W_1 = \mathbb{P}_{11} \cdot W_{11}$, where $W_{11} = (M+2c)^{2k} + 2M^{2k} + (M-2c)^{2k} - (M+a+b)^{2k} - (M+a-b)^{2k} - (M-a+b)^{2k} - (M-a-b)^{2k}$.

Since $a^2 \neq b^2$ we can use the binomial theorem to expand every term other than $2M^{2k}$ in W_{11} and get $W_{11} = 2M^{2k} + \sum_{i=0}^{2k}\binom{2k}{i}M^{2k-i}D_i$, where $D_i = (2c)^i + (-2c)^i - (a+b)^i - (a-b)^i - (-a+b)^i - (-a-b)^i$. Observe now that for odd $i, D_i = 0$. Moreover, we claim that $D_{2j} \geq 0$ for all $j \geq 1$. To see this claim observe that $(2a^2+2b^2) = (a+b)^2 + (a-b)^2$. We have $(2c)^{2j} = (2a^2+2b^2)^j = [(a+b)^2 + (a-b)^2]^j \geq (a+b)^{2j} + (a-b)^{2j}$ implying $W_{11} = 2M^{2k} + \sum_{j=0}^{k}\binom{2k}{2j}M^{2(k-j)}D_{2j} = \sum_{j=1}^{k}\binom{2k}{2j}M^{2(k-j)}D_{2j} \geq 0$.

Since $\mathbb{P}_{11} \geq 0$, we can get $W_1 \geq 0$.

(2) W_2 is the value under the condition of $|r_1| \neq |r_2|$.

For any $u, v \in \{-1,0,+1\}$ and $|u| \neq |v|$, we remember $\mathbb{P}(r_1 = u, r_2 = v)$ as $\mathbb{P}_{2uv}$. So, $W_2 = \sum_{u,v=-1}^{1}\mathbb{P}_{2uv}W_{2uv}$, where

$$\begin{aligned} W_{2uv} = &(M+uc+vc)^{2k} - (M+ua+vb)^{2k} \\ &+ (M+uc-vc)^{2k} - (M+ua-vb)^{2k} \\ &+ (M-uc+vc)^{2k} - (M-ua+vb)^{2k} \\ &+ (M-uc-vc)^{2k} - (M-ua-vb)^{2k}. \end{aligned}$$

Since $a^2 \neq b^2$ we can use the binomial theorem to expand every term in W_{2uv} and get $W_{2uv} = \sum_{i=0}^{2k}\binom{2k}{i}M^{2k-i}D_i$, where $D_i = (uc+vc)^i - (ua+vb)^i + (uc-vc)^i - (ua-vb)^i + (-uc+vc)^i - (-ua+vb)^i + (-uc-vc)^i - (-ua-vb)^i$. Observe now that for odd $i, D_i = 0$. So $W_{2uv} = \sum_{j=0}^{k}\binom{2k}{2j}M^{2(k-j)}D_{2j}$. The proof for the case is similar to the above. According to [17], we get $W_2 \geq 0$. To sum up, we get $W \geq 0$. □

Proof of Theorem 1

Proof For the proof of Theorem 1, we firstly prove the following inequalities:

For any $\varepsilon > 0$ and any unit vector $\mathbf{x} \in \mathbb{R}^d$,

$$\mathbb{P}[S(\mathbf{x}) > (1+\varepsilon)kd] < \exp\Big(-\frac{k}{2}(\varepsilon^2/2 - \varepsilon^3/3)\Big), \quad (9)$$

$$\mathbb{P}[S(\mathbf{x}) < (1-\varepsilon)kd] < \exp\Big(-\frac{k}{2}(\varepsilon^2/2 - \varepsilon^3/3)\Big). \quad (10)$$

We start with the upper tail.

Note that for an arbitrary vector $\mathbf{x}$, $S = S(\mathbf{x}) = \sum_{j=1}^{k}(\mathbf{x}\cdot\mathbf{c}_j)^2 = \sum_{j=1}^{k}Q_j^2(\mathbf{x})$. For arbitrary $h > 0$, according to Markov's inequality we get $\mathbb{P}\Big[S > (1+\varepsilon)kd\Big] = \mathbb{P}\Big[\exp(hS) > \exp\Big(h(1+\varepsilon)kd\Big)\Big] < \mathbb{E}\Big(\exp(hS)\Big)\exp\Big(-h(1+\varepsilon)kd\Big)$. Since $\{Q_j\}_{j=1}^{k}$ are i.i.d, we have: $\mathbb{E}(\exp(hS)) = \mathbb{E}\Big(\prod_{j=1}^{k}\exp(hQ_j^2)\Big) = \big(\mathbb{E}\left(\exp(hQ_1^2)\right)\big)^k$.

To optimize the bound, this gives $h = \frac{1}{2d}\cdot\frac{\varepsilon}{(1+\varepsilon)\log d} < \frac{1}{2d}$. Taking (4) to (11). Thus, for any $\varepsilon > 0$, one can see that

$$\begin{aligned} &\mathbb{P}\Big[S > (1+\varepsilon)kd\Big] \\ &< \Big(\mathbb{E}\left(\exp(hQ_1^2)\right)\Big)^k \exp\Big(-h(1+\varepsilon)kd\Big) \\ &< \Big(\frac{1}{\sqrt{1-2hd}}\Big)^k \exp\Big(-h(1+\varepsilon)kd\Big) \\ &= \Big(1 - \frac{\varepsilon}{(1+\varepsilon)\log d}\Big)^{-k/2}\exp\Big(\frac{-k\varepsilon}{2\log d}\Big). \end{aligned} \quad (11)$$

The proof of lower bound in (10) is similar to (9).

For arbitrary $h > 0$, we get that for any $\varepsilon > 0$, $\mathbb{P}\Big[S < (1-\varepsilon)kd\Big] = \mathbb{P}\Big[\exp(hS) < \exp\Big(h(1-\varepsilon)kd\Big)\Big] < \Big(\mathbb{E}\left(\exp(-hQ_1^2)\right)\Big)^k \exp\Big(h(1-\varepsilon)kd\Big)$. According to the distribution characteristics of $\mathbf{R}$ in (3), we know $\mathbb{E}(\mathbf{R}_{ij}) = \mathbb{E}(r_{ij}) = 0$, thus, we have $\mathbb{E}(r_{ij}^2) = \frac{(\pm 1)^2\times k}{d\times k} = \frac{1}{d}$ and $\mathbb{E}(Q_j) = \mathbb{E}\big(d\sum_{i=1}^{d}x_i r_{ij}\big) = d\mathbb{E}\big(\sum_{i=1}^{d}x_i r_{ij}\big) = d\sum_{i=1}^{d}x_i\mathbb{E}(r_{ij}) = 0$.

So, we obtain that

$$\begin{aligned} &\mathbb{E}(Q_j^2) = \mathbb{E}\left(\left(d\sum_{i=1}^{d}x_i r_{ij}\right)^2\right) \\ &= d^2\mathbb{E}\left(\left(\sum_{i=1}^{d}x_i r_{ij}\right)^2\right) \\ &= d^2\left[\sum_{i=1}^{d}x_i^2\mathbb{E}(r_{ij}^2) + \sum_{l=1}^{d}\sum_{m=1}^{d}2x_l x_m\mathbb{E}(r_{lj})\mathbb{E}(r_{mj})\right] \\ &= d\|x\|^2 = d \end{aligned} \quad (12)$$

Let us expand $\exp(-hQ_1^2)$ to get

$$\begin{aligned} &\mathbb{P}\Big[S(\mathbf{x}) < (1-\varepsilon)kd\Big] \\ &< \Big(\mathbb{E}\Big(1 - hQ_1^2 + \frac{(-hQ_1^2)^2}{2!}\Big)\Big)^k \exp\Big(h(1-\varepsilon)kd\Big) \\ &= \Big(1 - h\mathbb{E}(Q_1^2) + \frac{h^2}{2}\mathbb{E}(Q_1^4)\Big)^k \exp\Big(h(1-\varepsilon)kd\Big). \end{aligned} \quad (13)$$

Substituting (12) and (5) for (13), we get (15). Taking $h = \frac{1}{2d} \cdot \frac{\varepsilon}{(1+\varepsilon)\log d}$ which is not optimal but is still "good enough" and a series of expansion, we get (16):

$$\left(1 - h\mathbb{E}(Q_1^2) + \frac{h^2}{2}\mathbb{E}(Q_1^4)\right)^k \exp\left(h(1-\varepsilon)kd\right) \quad (14)$$

$$\leq \left(1 - hd + \frac{3}{2}(hd)^2\right)^k \exp\left(h(1-\varepsilon)kd\right) \quad (15)$$

$$\leq \left(1 - \frac{\varepsilon}{(1+\varepsilon)\log d}\right)^{-k/2} \exp\left(\frac{-k\varepsilon}{2\log d}\right). \quad (16)$$

Here, we complete the proof of upper and lower bounds.

As mentioned above, we have $\|f(\mathbf{x})\|^2 = S \times \frac{1}{kd}$. Let

$$2 \times \left(1 - \frac{\varepsilon}{(1+\varepsilon)\log d}\right)^{-k/2} \exp\left(\frac{-k\varepsilon}{2\log d}\right) \leq 2\delta/n^2,$$

we obtain

$$k_0 = \mathcal{O}\left((2\log(n) - \log\delta)\log d/\varepsilon\right).$$

Combining (9) and (10), for each of the $\binom{n}{2}$ pairs $\mathbf{u}, \mathbf{v} \in \mathbf{P}$, the squared norm of the vector $\mathbf{u} - \mathbf{v}$, is maintained within a factor of $1 \pm \varepsilon$. Therefore, if for some family r_{ij} as (3) we can get that for some $\delta > 0$ and any fixed vector $\mathbf{x} \in \mathbb{R}^d$, $\mathbb{P}[(1-\varepsilon)\|x\|^2 \leq \|f(x)\|^2 \leq (1+\varepsilon)\|x\|^2] \geq 1 - 2\delta/n^2$, then the probability of not getting right results is bounded by $\binom{n}{2} \times 2\delta/n^2 < \delta$. □

VI. Conclusions

Theoretical analysis shows that the proposed ESE has the minimum time complexity $\mathcal{O}\left(\varepsilon^{-1}n\log(n)\log(d)\right)$ in fast matrix multiplication and keeps the best accuracy $(1+\varepsilon)$ for the general case, compared with the state-of-the-art embedded algorithms. Experimental results demonstrate that ESE has a significant advantage over time complexity than other dimension reduction algorithms with satisfactory accuracy.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (No.61703396, No.61673293, No.62076234), the CCF-Tencent Open Fund, the Youth Innovation Promotion Association CAS, the Excellent Talent Introduction of Institute of Information Engineering of CAS (No. Y7Z0111107), Beijing Outstanding Young Scientist Program (NO. BJJWZYJH012019100020098), and the Beijing Municipal Science and Technology Project (No. Z191100007119002).

References

[1] F. Fang and Z. Yu, "Model averaging assisted sufficient dimension reduction," *Computational Statistics & Data Analysis*, p. 106993, 2020.

[2] J. Xi, T. Zhao, Q. Li, B. Wang, X. Wang, and X. Zhan, "The research on feature extraction method of ecg signal based on kpca dimension reduction," in *Proceedings of the 2020 12th International Conference on Machine Learning and Computing*, 2020, pp. 500–504.

[3] L. M. Abualigah, A. T. Khader, M. A. Al-Betar, and O. A. Alomari, "Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering," *Expert Systems with Applications*, vol. 84, pp. 24–36, 2017.

[4] M. Juuti, F. Corona, and J. Karhunen, "Stochastic discriminant analysis for linear supervised dimension reduction," *Neurocomputing*, vol. 291, pp. 136–150, 2018.

[5] Y. Yang, Q. J. Wu, and Y. Wang, "Autoencoder with invertible functions for dimension reduction and image reconstruction," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 7, pp. 1065–1079, 2016.

[6] R. Yin, Y. Liu, W. Wang, and D. Meng, "Sketch kernel ridge regression using circulant matrix: Algorithm and theory," *IEEE transactions on neural networks and learning systems*, 2019.

[7] R. Yin, Y. Liu, L. Lu, W. Wang, and D. Meng, "Divide-and-conquer learning with nyström: Optimal rate and algorithm." in *AAAI*, 2020, pp. 6696–6703.

[8] Z. Tang, Y. Shen, K. Zhou, K. Cao, F. Yang, D. Cai, and C. Zhou, "A dimension reduction method used in detecting errors of distribution transformer connectivity," *IEEE Access*, vol. 8, pp. 79 408–79 418, 2020.

[9] A. Naor, G. Pisier, and G. Schechtman, "Impossibility of dimension reduction in the nuclear norm," *Discrete & Computational Geometry*, vol. 63, no. 2, pp. 319–345, 2020.

[10] Z. Cai, R. Li, and L. Zhu, "Online sufficient dimension reduction through sliced inverse regression," *Journal of Machine Learning Research*, vol. 21, no. 10, pp. 1–25, 2020.

[11] O. Zahm, P. G. Constantine, C. Prieur, and Y. M. Marzouk, "Gradient-based dimension reduction of multivariate vector-valued functions," *SIAM Journal on Scientific Computing*, vol. 42, no. 1, pp. A534–A558, 2020.

[12] F. W. Townes, S. C. Hicks, M. J. Aryee, and R. A. Irizarry, "Feature selection and dimension reduction for single-cell rna-seq based on a multinomial model," *Genome biology*, vol. 20, no. 1, pp. 1–16, 2019.

[13] R. Pasunuri, V. C. Venkaiah, and B. Dhariyal, "Ascending and descending order of random projections: Comparative analysis of high-dimensional data clustering," in *Harmony Search and Nature Inspired Optimization Algorithms*. Springer, 2019, pp. 133–142.

[14] X.-D. Wang, R.-C. Chen, Z.-Q. Zeng, C.-Q. Hong, and F. Yan, "Robust dimension reduction for clustering with local adaptive learning," *IEEE transactions on neural networks and learning systems*, vol. PP, no. 99, pp. 1–13, 2018.

[15] C. Boutsidis, A. Zouzias, M. W. Mahoney, and P. Drineas, "Randomized dimensionality reduction for k-means clustering," *IEEE Transactions on Information Theory*, vol. 61, no. 2, pp. 1045–1062, 2015.

[16] C. Boutsidis, M. W. Mahoney, and P. Drineas, "Unsupervised feature selection for the k-means clustering problem," in *Advances in Neural Information Processing Systems*, 2009, pp. 153–161.

[17] D. Achlioptas, "Database-friendly random projections: Johnson-lindenstrauss with binary coins," *Journal of computer and System Sciences*, vol. 66, no. 4, pp. 671–687, 2003.

[18] P. Li, T. J. Hastie, and K. W. Church, "Very sparse random projections," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 287–296.

[19] W. Liu, X. Shen, and I. Tsang, "Sparse embedded k-means clustering," in *Advances in Neural Information Processing Systems*, 2017, pp. 3321–3329.

[20] K. Makarychev, Y. Makarychev, and I. Razenshteyn, "Performance of johnson-lindenstrauss transform for k-means and k-medians clustering," *arXiv preprint arXiv:1811.03195*, 2018.

[21] A. Naor, "Metric dimension reduction: A snapshot of the ribe program," *arXiv preprint arXiv:1809.02376*, 2018.

[22] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay, "Clustering in large graphs and matrices." in *Tenth Acm-siam Symposium on Discrete Algorithms*, 1999.

[23] D. Feldman, M. Schmidt, and C. Sohler, "Turning big data into tiny data: Constant-size coresets for k-means, pca and projective clustering," in *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 2013, pp. 1434–1453.

[24] D. M. Kane and J. Nelson, "Sparser johnson-lindenstrauss transforms," *Journal of the Acm*, vol. 61, no. 1, pp. 1–23, 2014.

[25] F. Pourkamali-Anaraki and S. Becker, "Preconditioned data sparsification for big data with applications to pca and k-means," *IEEE Transactions on Information Theory*, vol. 63, no. 5, pp. 2954–2974, 2017.

[26] W. B. Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a hilbert space," *Contemporary Mathematics*, vol. 26, 1984.

[27] R. I. Arriaga and S. Vempala, "An algorithmic theory of learning: Robust concepts and random projection," *Machine Learning*, vol. 63, no. 2, pp. 161–182, 2006.