Random Projections for Non-linear Dimensionality Reduction

Long Cheng, Chenyu You, and Yani Guan

Abstract—**The need to analyze high-dimensional data in various areas, such as image processing, human gene regulation and smart grids, raises the importance of dimensionality reduction. While classical linear dimensionality reduction methods are easily implementable and efficiently computable, they fail to discover the true structure of high-dimensional data lying on a non-linear subspace. To overcome this issue, many non-linear dimensionality reduction approaches, such as Locally Linear Embedding, Isometric Embedding and Semidefinite Embedding, have been proposed. Though these approaches can learn the global structure of non-linear manifolds, they are computationally expensive, potentially limiting their use in large-scale applications involve very high-dimensional data. An innovative method framework that combines random projections and non-linear dimensionality reduction methods is proposed in this paper to increase computational speed and reduce memory usage, while preserving the non-linear data geometry. Illustrations with various combinations of random projections and non-linear dimensionality reduction methods tested on a hand-written digits dataset are also given in this paper to demonstrate that this method framework is both computationally efficient and globally optimal.**

Index Terms—**Random projections, nonlinear dimensionality reduction, manifold learning, high dimensional data.**

I. Introduction

The rapid increase in the size of data generated by acquisition systems poses a significant challenge to analyze and process data with high efficiency. This is particularly evident in cases of image [1], smart grids [2], biological data [3] and education statistics [4], where high-dimensional representations are inevitable. However, in most cases, these high-dimensional representations are correlated in a certain fashion, and contain too much redundant information. In order to overcome this defect and enhance efficiency, using a small number of dimensions to effectively model the high dimensional data with minimal information loss has drawn more and more attention.

The goal of dimensionality reduction is to find a $k-$ dimensional representation of a $n-$ dimensional vector $\mathbf{x}$, with $k \ll n$. Assuming real-valued quantities, this process can be mathematically formalized as finding a mapping ∅ such that ∅: $\mathbb{R}^n \mapsto \mathbb{R}^k$. A straightforward approach to finding this mapping is to compute $\mathbf{w} \in \mathbb{R}^{n \times k}$ such that $\hat{\mathbf{x}} = \mathbf{w}^{\mathrm{T}}\mathbf{x} \in \mathbb{R}^k$. This is the general linear dimensionality reduction framework. Of the various methods proposed to compute this mapping matrix $\mathbf{w}$, principal components analysis (PCA) [5] and linear discriminant analysis (LDA) [6] are the most popular. Other techniques include factor analysis [7], independent component analysis [8] and projection pursuit [9].

Linear dimensionality reduction techniques work well if the data is confined to an underlying low-dimensional subspace. However, if data are actually sampled from an underlying non-linear manifold, these techniques cannot be applied in principle. To overcome this issue, several non-linear dimensionality reduction techniques have been proposed, of which we briefly review the most popular ones. Locally Linear Embedding (LLE) [10] is based on the intuition that non-linear data can be modelled locally linearly, i.e., it exploits local neighborhood information to discover the underlying global structure of the data. IsoMAP [11] exploits the intuition that Euclidean distance may not be an appropriate distance metric to quantify the similarity between data points that lie on a manifold, preserving the ``true" structure and the global properties of the data through geodesic distances in a neighborhood graph. Semidefinite embedding (SDE) [12] enforces constraints on the distances between points and angles between edges in a neighborhood graph, ensuring that neighbors of points are preserved in the embedding.

The non-linear dimensionality reduction techniques discussed in the previous paragraph have demonstrated impressive results in recovering complex non-linear manifolds. However, these methods are extremely computationally expensive, potentially limiting their use in large-scale applications involving very high-dimensional data. Methods such as LLE and Isomap rely on computing nearest neighbors for each data point, which can be prohibitively compute-intensive for high-dimensional data. SDE involves solving an optimization problem that belongs to a class of problems called semidefinite programming, which simply does not scale to large datasets. Therefore, we see that there is an immediate requirement to come up with faster non-linear dimensionality reduction techniques.

Random projections (RP) belong to a class of linear dimensionality reduction techniques that employ randomized constructions for the mapping matrix $\mathbf{w}$. Their applicability to diverse application areas is in part due to their ability to preserve all pairwise distances between the data points with high probability in the low-dimensional space [13]. An

Manuscript received June 20, 2016; revised August 4, 2016.

Long Cheng is with the Kiwii Power Technology Co., Ltd, Troy, NY, USA (e-mail: dearlongcheng@gmail.com).

Chenyu You and Yani Guan are with the Rensselaer Polytechnic Institute, Troy, NY, USA (e-mail: yani_guan@kiwiipower.com, charlesyou999648@gmail.com).

doi: 10.18178/ijmlc.2016.6.4.601

immediate idea would be to combine RP with non-linear dimensionality reduction techniques to reduce the computational resources required. The general framework we will work in is the following: given the data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ of m points in an $n-$ dimensional space, we will first perform linear dimensionality reduction as $\hat{\mathbf{X}} = \mathbf{X} \bullet \boldsymbol{R} \in \mathbb{R}^{\boldsymbol{m} \times \boldsymbol{k}}$, where $\boldsymbol{R} \in \mathbb{R}^{\boldsymbol{n} \times \boldsymbol{k}}$ can be any randomized construction, such as the random Gaussian matrix or the random Sign matrix. We will then apply the usual non-linear dimensionality techniques on the matrix $\hat{\mathrm{X}}$. The advantage now is that we will be working in the $k-$dimensional space. With $k \ll n$ typically, this amounts to a significant reduction in computational complexity.

II. Related Work

The use of random projections to perform non-linear dimensionality reduction has been explored in prior work. Baraniuk and Wakin [14] provided provable guarantees that a small number of random projections can preserve all pairwise distances between points sampled from an underlying non-linear manifold. Hegde *et al.* [15] extended this result to estimate the structure of the manifold. They also proposed an algorithm to estimate the required dimension k. Freund *et al.*[16] proposed a new data structure, which they called the RP tree to learn non-linear manifold structure. While this method also does not directly combine random projections and non-linear dimensionality reduction, an interesting aspect of RP trees is that instead of considering all data points and their neighbors to form a connected graph, as done in Isomap, here a top-down approach is employed. That is, it starts with the entire dataset and partitions it into smaller regions in a hierarchical fashion. Data points in these smaller regions are now approximated using affine subspaces, thereby avoiding the large computation time associated with nearest neighbor search.

The work reported in Chui and Wang [17] is perhaps the closest to what we wish to do with random projections and non-linear dimensionality reduction. They proposed combining random projections and anisotropic tranformations with Isomap to reduce the associated computational complexity. Raducanu and Dornaika [18] proposed a supervised version of the Laplacian Eigenmaps (LE) [19]. While not being the main focus of their paper, they performed random projections prior to learning the underlying structure of the manifold to reduce the computational complexity of their overall system. Similarly, Talwalkar *et al.* [20] also used random projections to compute approximate nearest neighbors prior to constructing the neighborhood graph. Again, the main focus of their work was on developing approximate spectral decomposition methods that speed up manifold learning using Isomap and Laplacian Eigenmaps. An interesting aspect of this work was that empirical results were demonstrated on datasets consisting of millions of face images, providing a very good testbed to study the limitations of the existing approaches. Sun *et al.* [21] also used random projections to reduce feature space dimensionality as part of a fast nearest neighbor search technique to speed up the local tangent space alignment (LTSA) [22] technique for non-linear dimensionality reduction.

Finally, we note that while there has been some work in combining random projections with existing non-linear dimensionality methods, as discussed above, most of this work is ad-hoc in the sense that there are no provable theoretical results that explain why using random projections prior to performing non-linear dimensionality reduction gives similar or better results than performing dimensionality reduction in the original feature space directly.

III. Method Framework

A. Random Projections

A general framework of performing dimensionality reduction is the following: given the data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ of m points in an $n-$dimensional space, we can first use a matrix $\boldsymbol{R} \in \mathbb{R}^{\boldsymbol{n} \times \boldsymbol{k}}$ to reduce the dimensionality of the rows of $\mathbf{X}$ as follows:

$$\hat{\mathbf{X}} = \mathbf{X} \bullet \boldsymbol{R} \in \mathbb{R}^{\boldsymbol{m} \times \boldsymbol{k}} \tag{1}$$

The idea behind random projections arises from the Johnson-Lindenstrauses lemma [23]: if data in a vector space are projected onto a randomly chosen high-dimensional subspace, then the pairwise distances between the original data are approximately preserved. Such an $\boldsymbol{R}$ matrix can be constructed by setting its entries to be independent Gaussian random variables. We also note that the computation of $\mathbf{X} \bullet \boldsymbol{R}$, in general, takes $o(mnk)$ time, which could be very large in high dimensional data settings. And if the data matrix $\boldsymbol{R}$ is sparse with $\boldsymbol{r}$ nonzero entries per column, the complexity is of order $o(rmk)$. So it is of particular interest to improve the running time by carefully constructing the matrix $\boldsymbol{R} \in \mathbb{R}^{\boldsymbol{n} \times \boldsymbol{k}}$.

An even simpler construction of $\boldsymbol{R}$ can be achieved by setting R_{ij} to $1/\sqrt{k}$ with probability $1/2$ or to $-1/\sqrt{k}$ with probability $1/2$ for all i and j. This is called a random sign matrix. Dimitris Achlioptas further proposed a sparse version of the random sign matrix. The entries of $\boldsymbol{R}$ are set to:

$$R_{ij} = \sqrt{\frac{3}{k}} \times \begin{cases} +1 & ,\text{w.p. } 1/6 \\ 0 & ,\text{w.p. } 2/3 \\ -1 & ,\text{w.p. } 1/6 \end{cases} \tag{2}$$

It can be seen that $\boldsymbol{R}$ has about $2/3$ of its entries equal to zero, making the computation of the product $\mathbf{X} \bullet \boldsymbol{R}$ faster and easier.

B. Non-linear Dimensionality Reduction

Non-linear dimensionality reduction methods are developed in order to find an embedding of the data in a $k-$dimensional space with $k \ll n$ to capture the intrinsic structure of the original dataset in a very non-linear situation, where linear dimensionality reduction methods fail to produce accurate results.

1) Locally Linear Embedding:

LLE introduces the idea of using linear regression to reconstruct a point from its neighbors. For each point $i = 1, 2, 3,\ldots, m$, let Γ_i denote the set of neighbors of this point. The simplest formulation of LLE, which is to find the K nearest neighbors of the $i-th$ point as measured by Euclidean distance, is described as follows:

$$\min_{w_{i1}, w_{i2}, \ldots w_{iK}} \left\| X_{i*} - \sum_{j \in \Gamma_i} w_{ij} X_{j*} \right\|_2^2 \tag{3}$$

The above optimization problem is solved for all points $i = 1, 2, 3,\ldots, m$, which means we need to minimize the following objective:

$$\sum_{i=1}^{m} \left\| X_{i*} - \sum_{j \in \Gamma_i} w_{ij} X_{j*} \right\|_2^2 \tag{4}$$

over all possible choices for w_{ij}. And $\sum_{j \in \Gamma_i} w_{ij} = 1$ should be satisfied. To find a global embedding of all m points in a lower-dimensional space, k dimensional vectors $\boldsymbol{y_i} \in \mathbb{R}^{\boldsymbol{k}}$ for $i = 1, 2, 3,\ldots, m$, are chosen to minimize the following embedding cost function:

$$\sum_{i=1}^{m} \left\| y - \sum_{j \in \Gamma_i} w_{ij} j \right\|_2^2 \tag{5}$$

over all possible choices for vectors $\boldsymbol{y_i} \in \mathbb{R}^{\boldsymbol{k}}$ for all $i = 1, 2, 3,\ldots, m$. Extra constraints as follows are necessary to make the Equation (5) well-posed and can be minimized efficiently:

$$\sum_{i=1}^{m} y_i = \mathbf{0} \tag{6}$$

$$\sum_{i=1}^{m} y_i y_i^T = m\mathbf{I}_k \tag{7}$$

where $\mathbf{I}_k$ is the $k \times k$ identity matrix. Solving the above optimization problem turns out to be an SVD computation.

2) IsoMAP:

Given a set of points in a high dimensional space, Euclidean distances between different points are easy to compute but might fail to capture the intrinsic geometric structure of non-linear manifolds. IsoMAP can be used to avoid this problem by constructing a neighborhood graph. This can be achieved by either connecting points to their K nearest neighbors, or connecting points that are smaller than a threshold value ε, where the distance between points is the Euclidean distance. Then the point-to-point distance can be calculated as the length of the shortest path between points in the neighborhood graph. Thus an $m \times m$ matrix, the (i, j)-th entry of which is the length of the shortest path between points i and j in the neighborhood graph, can be formed. At last, the SVD is applied on this matrix to determine the top k left singular vectors as the low-dimensional projection of the m points in a $k-$ dimensional space.

3) Semidefinite Embedding:

Let $y_i \in \mathbb{R}^k$ denote the embedding of original points in the $k-$dimensional space. For all $n-$dimensional points X_{i*}, $i = 1, 2, 3,\ldots, m$, neighborhoods are formed to preserve distances and angles within those neighborhoods, the constrains of which can be written as follows:

$$\left\| A_{i*} - A_{j*} \right\|_2^2 = \left\| y_i - y_j \right\|_2^2 \tag{8}$$

for all pairs of points that are neighbors of each other or have a common neighbor. Another mean-centering constraint to eliminate a translational degree of freedom in the embedding is as follows:

$$\sum_{i=1}^{m} y_i = 0 \tag{9}$$

At last, SDE attempts to “unfold” the points y_i as far apart from each other as possible, which equals to maximize the following function:

$$\sum_{all \;\; pairs\,(i,j)} \left\| y_i - y_j \right\|_2^2 \tag{10}$$

This is a convex problem and can be solved efficiently. But obviously it is more computationally expensive than LLE and IsoMAP, which compute SVD in the end.

C. Dimensionality Reduction Method Framework based on Random Projections and Non-linear Dimensionality Reduction

Though random projections are very efficient in terms of computational time and implementation, they could not preserve as much information as those non-linear dimensionality reduction methods if the high-dimensional data are sampled from an underlying non-linear manifold. Non-linear dimensionality reduction methods, on the other hand, are computationally expensive, but can capture the intrinsic structure of the original dataset more accurately. Therefore, a hybrid dimensionality reduction method framework that combines the advantages of the two approaches is suitably proposed in this paper. The framework works as follows: data are first pre-processed with random projections to a lower dimension k_1 with $k_1 << n$; then non-linear dimensionality reduction methods are further applied on the k_1- dimensional data to reduce its dimensionality to k_2 with $k_2 < k_1$. This framework intuitively will improve accuracy for random projections and running time efficiency for non-linear dimensionality reduction methods.

IV. Experimental Results

A. Evaluation Protocol and Datasets

We empirically evaluate the impact of combining random projections with three non-linear dimensionality reduction techniques: LLE, Isomap, and Laplacian Eigenmaps [14]. We compare the downstream classification accuracy of this approach with the baseline approach of performing dimensionality reduction in the original $n-$ dimensional feature space. In all our experiments, we work with 2313 images selected from handwritten zip-code digit images (300 pixels/inch in 8-bit gray scale) [24], consisting of the digits **1**, **6**, and **9.** The "average" one, the "average" six and the "average" nine are computed and plotted in Fig. 1. We use the $k-$nearest neighbors classifier to compute the classification accuracy, which is averaged over 20 randomly chosen training and testing splits. We use the leave one out cross-validation mechanism to pick the value for k.

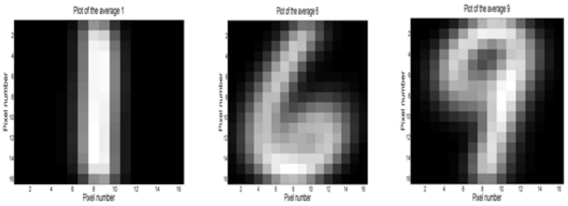

Fig. 1. Plots of the "average" one, six and nine.

B. Results

We begin with results using LLE. Prior to performing LLE, we project the original $n-$dimensional data (in the case of the digits data, $n = 256$) to a $k-$ dimensional space. We perform experiments with values of k ranging from 5 to 100. For the random projection matrix, we choose the random Gaussian matrix, the random sign matrix, and the sparse random sign matrix. In all experiments, LLE is used to reduce the input feature space to $\hat{d} = 3$ dimensions. We note that since the goal here is to obtain an extensive performance comparison between applying non-linear dimensionality reduction with and without randomly projected features, we did not perform any cross-validation to pick the value of $\hat{d}$ and chose it arbitrarily. The average classification results obtained are plotted in Fig. 2. And we note that in this plot, the baseline performance, *i.e.*, for each value of k, we randomly choose 20 different training and testing splits and compute the average performance. The goal is to compare the performance of randomly projected features with the baseline while using the same training and testing splits. We make the following observations:

1) The performance obtained by each of the random projection matrices approximately saturates beyond a certain projection dimension. This saturation performance is close to that of the baseline performance. In fact, for some values of k , the performance of randomly projected features is slightly better than the baseline performance.
2) There does not seem to be any significant differences in the performance of the three random projection matrices.

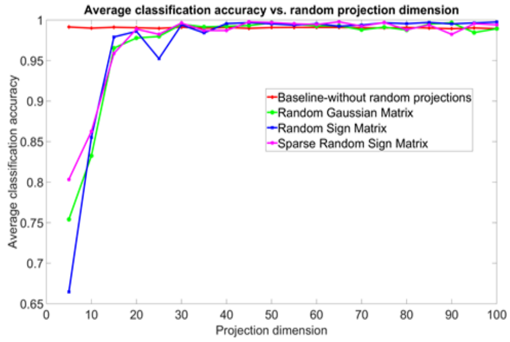

Fig. 2. A plot comparing the baseline classification accuracy with that obtained by combining random projections with LLE.

We next perform a running time comparison of performing LLE in the original feature space and the randomly projected feature space. All running times reported here are computed using MATLAB's in-built stopwatch timer on an Intel Xeon CPU with 16 GB RAM. A plot of the running time (averaged over 20 trials) in each case versus the random projection dimension is shown in Fig. 3. Here, we use all the data points to perform dimensionality reduction and do not consider any training and testing splits since the goal is to study the running time behavior of the two different approaches. Clearly, we see that the average running time of performing LLE in the randomly projected feature space is smaller than that of the original feature space.

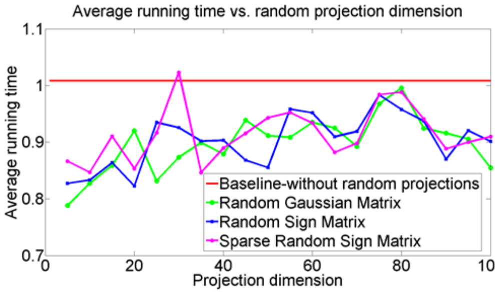

Fig. 3. A plot comparing the baseline running time (in seconds) with that obtained by combining random projections with LLE.

Additionally, we notice that the three different random projection methods seem to have almost similar average running times. These results, combined with the average accuracy results discussed above, clearly demonstrate the impact of using random projections prior to performing non-linear dimensionality reduction.

We next repeat the same experiments as described above with two other non-linear dimensionality reduction techniques: Isomap and Laplacian Eigenmaps. The average classification performance comparison plot for the two techniques are shown in Fig. 4 and Fig. 5.

The corresponding average running time comparison plots are shown in Fig. 6 and Fig. 7. Clearly, we observe a similar performance and running time trend with these two techniques as well.

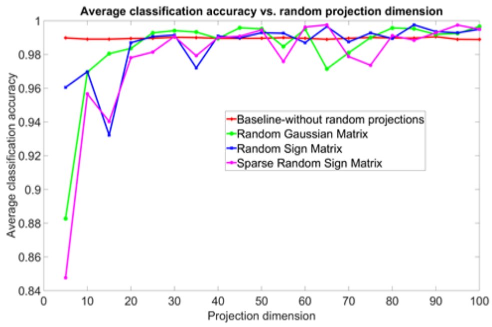

Fig. 4. A plot comparing the baseline classification accuracy with that obtained by combining random projections with Isomap.

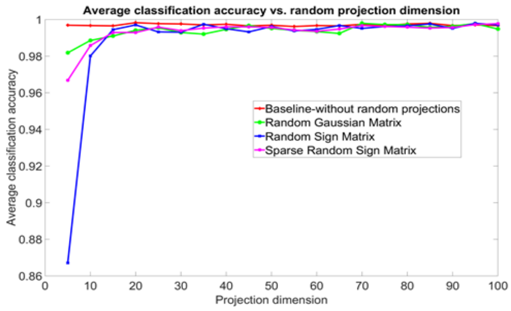

Fig. 5. A plot comparing the baseline classification accuracy with that obtained by combining random projections with Laplacian Eigenmaps.

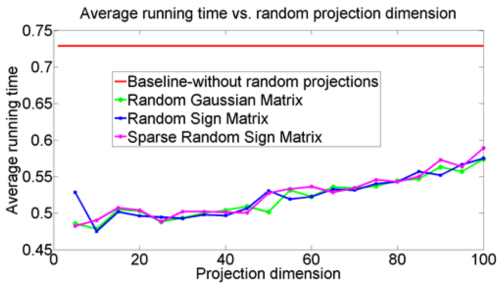

Fig. 6. A plot comparing the baseline running time (in seconds) with that obtained by combining random projections with Isomap.

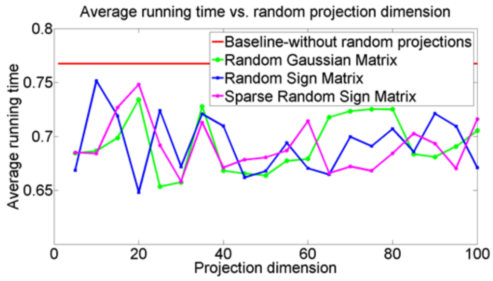

Fig. 7. A plot comparing the baseline running time (in seconds) with that obtained by combining random projections with Laplacian Eigenmaps.

V. Conclusions and Future Work

In this work, we considered the problem of combining random projections with non-linear dimensionality reduction techniques. Specifically, we randomly projected the input features to a lower dimensional space prior to applying non-linear dimensionality reduction. We observed empirically that the classification performance of the feature mapping learnt by applying non-linear dimensionality reduction after randomly projecting the input features is very close to, and in some cases, better than that learnt without performing random projections. Additionally, we also observed empirically that performing non-linear dimensionality reduction in the randomly projected feature space is faster than that in the original feature space. The framework proposed in this paper can be widely used in speech and image processing, smart grids, human gene distribution and so on.

As noted earlier, while empirical results demonstrate the efficacy of the proposed framework combining random projections with non-linear dimensionality reduction techniques in the manner as done in this work, establishing provable results justifying this observation is one line of work we wish to explore in the future.

In addition, we also find out that the efficiency of this framework is also closely related with the original structure of the high-dimensional dataset. For sparse high-dimensional dataset, the sparseness property of the original dataset will probably be lost after performing random projections, which will result in even lower running time efficiency. That is because the sparseness of the original dataset makes it possible to use some efficient algorithms in the process of non-linear dimensionality reduction, such as the less costly SVD algorithm suitable for sparse matrix. On the other hand, the loss of the sparse format of the original dataset will consume more memories to store the dense matrix and perform the decomposition after random projections are applied on the original sparse dataset. We will discuss this topic in our another paper that to be published in the near future.

References

[1] X. Desquesnes, A. Elmoataz, O. L ézoray, and V. T. Ta, "Efficient algorithms for image and high dimensional data processing using eikonal equation on graphs," *Advances in Visual Computing*, Springer Berlin Heidelberg, pp. 647-658, 2010.

[2] L. Cheng, L. Wang, and F. Gao, "Power system fault classification method based on sparse representation and random dimensionality reduction projection," in *Proc. IEEE Power & Energy Society General Meeting*, 2015, pp. 1-5.

[3] R. Clarke, H. W. Ressom, A. Wang, J. Xuan, M. C. Liu, E. A. Gehan, and Y. Wang, "The properties of high-dimensional data spaces: implications for exploring gene and protein expression data," *Nature Reviews Cancer*, vol. 8, no. 1, pp. 37-49, 2008.

[4] L. Cheng and C. You, "Analysis of rising tuition rates in the United States based on clustering analysis and regression models," in *Proc. Conference in Computer Science & Information Technology*, 2016, pp. 127-144.

[5] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37-52, 1987.

[6] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2012, John Wiley & Sons.

[7] L. L. Thurstone, *Multiple Factor Analysis*, University of Chicago Press, 1947.

[8] A. Hyv ärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, 2004, vol. 46, John Wiley & Sons.

[9] P. J. Huber, "Projection pursuit," *The Annals of Statistics*, pp. 435-475, 1985.

[10] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323-2326, 2000.

[11] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319-2323, 2000.

[12] K. Q. Weinberger and L. K. Saul, "Unsupervised learning of image manifolds by semidefinite programming," *International Journal of Computer Vision*, vol. 70, no. 1, pp. 77-90, 2006.

[13] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proc. 13th Annual ACM Symposium on Theory of Computing*, 1998, pp. 604-613.

[14] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction," *Neural Computation*, vol. 15, no. 6, pp. 1373-1396, 2003.

[15] C. Hegde, M. Wakin, and R. Baraniuk, "Random projections for manifold learning," *Advances in Neural Information Processing Systems*, pp. 641-648, 2008.

[16] Y. Freund, S. Dasgupta, M. Kabra, and N. Verma, "Learning the structure of manifolds using random projections," *Advances in Neural Information Processing Systems*, pp. 473-480, 2007.

[17] R. G. Baraniuk and M. B. Wakin, "Random projections of smooth manifolds," *Foundations of Computational Mathematics*, vol. 9, no. 1, pp. 51-77, 2009.

[18] B. Raducanu and F. Dornaika, "A supervised non-linear dimensionality reduction approach for manifold learning," *Pattern Recognition*, vol. 45, no. 6, pp. 2432-2444, 2012.

[19] C. K. Chui and J. Wang, "Randomized anisotropic transform for nonlinear," *GEM-International Journal on Geo-*, vol. 1, no. 1, pp. 23-50, 2010.

[20] A. Talwalkar, S. Kumar, and H. Rowley, "Large-scale manifold learning," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.

[21] W. Sun, A. Halevy, J. J. Benedetto, W. Czaja, W. Li, C. Liu, B. Shi, and R. Wang, "Nonlinear dimensionality reduction via the ENH-LTSA method for hyperspectral image classification," *Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 2, pp. 375-388, 2014.

[22] Z. Zhang and H. Zha, "Nonlinear dimension reduction via local tangent space alignment," *Intelligent Data Engineering and Automated Learning*, Springer Berlin Heidelberg, pp. 477-481, 2003.

[23] W. B. Johnson and J. Lindenstrauss, "Extensions of lipschitz mapping into hilbert space," in *Proc. Conference in Modern Analysis and Probability*, 1984, vol. 26, pp. 189-206.

[24] J. J. Hull, "A database for handwritten text recognition research," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 550-554, 1994.

Long Cheng was born in Yuncheng, Shanxi, China, on November 29th 1991. He received his Master's degrees in both electrical engineering and applied mathematics from Rensselaer Polytechnic Institute, Troy, NY, USA in May 2015 and his B.S. in electrical engineering and automation from Tianjin University, Tianjin, China in July 2013.

He is currently working as the COO and Research Scientist at Kiwii Power Technology Co., Ltd, Troy, NY, USA. Before that, he worked as a Data Scientist at Rang Technologies Inc, Piscataway, NJ, USA. He also previously interned at IBM Research China. His research interests include machine learning, data mining, signal processing and smart grids.

Chenyu You was born in Wuxi, Jiangsu, China, on September 15th 1993. He is currently pursuing his B.S. in electrical engineering with a minor in mathematics from Rensselaer Polytechnic Institute, Troy, NY, USA. His research interests are machine learning, data mining, statistical signal processing and mathematical modelling.

Yani Guan was born in Jiaxing, Zhejiang, China, on October 8th 1993. She is pursuing her B.S. in electrical engineering with a minor in psychology at Rensselaer Polytechnic Institute, Troy, NY, USA. She is currently working as the CEO at Kiwii Power Technology Co., Ltd. Her research interests are machine learning, neural networks and radio frequency.