

# Principal Component Analysis Approach in Selecting Type-1 and Type-2 Fuzzy Membership Functions for High-dimensional Data

Desh Raj<sup>\*†</sup>, Aditya Gupta<sup>†‡</sup>, Kenil Tanna<sup>\*‡</sup>, Bhuvnesh Garg<sup>†‡</sup>, and Frank Chung-Hoon Rhee<sup>‡</sup>, *Member, IEEE*

<sup>\*</sup>Department of Computer Science and Engineering

Indian Institute of Technology Guwahati, Guwahati, India 781039

Email: {r.desh,kenil}@iitg.ac.in

<sup>†</sup>Department of Mathematics

Indian Institute of Technology Guwahati, Guwahati, India 781039

Email: {aditya.gupta,g.umang}@iitg.ac.in

<sup>‡</sup>Department of Electronics and Communication Engineering

Hanyang University, Ansan-Si, Korea 15588

Email: frhee@fuzzy.hanyang.ac.kr, Contact: +82.31.400.5296, Fax: +82.31.436.8152

**Abstract**—With increased interest in learning from data, algorithms that manipulate datasets containing hundreds of features have become popular in various fields such as medicine, image processing, geolocation, biochemistry, and computational linguistics. Since a number of these applications exploit the power of fuzzy sets in representing uncertainties, it may be considered essential to describe a method for selecting the most suitable fuzzy membership function to represent a high-dimensional dataset. In this paper, we propose such a method, which is based on dimensionality reduction using the Principal Component Analysis (PCA) technique, followed by the Wilcoxon Minimal Bin Size algorithm, which has earlier been evaluated on multidimensional datasets up to 8 dimensions. We further demonstrate our proposed method using two real datasets consisting of 281 and 500 features, respectively.

## I. INTRODUCTION

In the last decade, especially with the explosion of deep learning methods in computer vision and natural language processing, computing data with hundreds (or thousands) of features has become a popular technique in information extraction. While the conventional machine learning methods required rigorous feature engineering in the form of manual selection of attributes from a dataset, several layered neural networks are now being used to model high level abstractions from data.

More attributes tend to capture those characteristics of a dataset that may not be easily observable for manual selection. Further, if the width (number of features) of a dataset is comparable to (or larger than) its length (number of instances), it gives rise to some interesting results. First, certain features may be some linear or non-linear combination of the values of a subset of the other features, so that these features may be deconvolved from the high-dimensional data [1]. This property forms the basis for Principal Component Analysis (PCA) and other dimensionality reduction techniques. Second, various mixture modeling techniques such as the hierarchical

model [2] assume that a hypothesis test is performed for each observation in which case it is possible to use the known null distribution to estimate the common probability that an observation is drawn from the null. Third, the empirical Bayes variance shrinkage method [3] calculates the log ratios of the mean and variance for each feature by assuming a common linear distribution across all features. The hyperparameters for this distribution are then determined and are used for adjusting any individual measurement's estimates. Fourth, the surrogate variable analysis [4] exploits a supervised PCA approach for determining pairs of surrogate and confounder variables by assuming that each observation is influenced by a single variable of interest.

Applications of fuzzy logic systems (FLSs) have been extended to fields such as control of mobile robots [5], forecasting of time-series [6], pre-processing radiographic images [7], and numerous others, most of which require high-dimensional data. With the availability of the different types of fuzzy membership functions (MFs), the user may be faced with a predicament of which MF would best represent the data set for the particular example. Ongoing studies to reduce computational complexity for type-2 (T2) FLSs may prove important for shifting fuzzy models from type-1 (T1) to T2 or higher levels of fuzziness. However, as described by Wu [8] and Mendel [9], T2 FLSs do not always outperform T1 FLSs, and it may be advantageous to identify applications where T1 FLSs suffice, given the uncertainty involved.

For these reasons, Raj et al. [10] recently proposed an iterative algorithm which employs the Wilcoxon Signed Rank (WSR) and Wilcoxon Rank Sum (WRS) tests on non-parametric data, and returns the smallest bin size (i.e., highest precision level) for which two data sets may be considered similar. This similarity score may then be used to determine an appropriate fuzzy MF to be used for the particular dataset. We refer to this method as the “Wilcoxon Minimal Bin Size

(WMBS)” algorithm in the remaining text.

As with most pattern recognition techniques, the WMBS algorithm suffers from the curse of dimensionality due to the fact that the number of bins increases exponentially with the number of dimensions. Further, the requirement of simultaneous access to the entire matrix of bins for the generation of the “snaky” asymmetric Gaussian fuzzy MF [11] discards the possibility of sequential access. Therefore, even though the algorithm may be theoretically robust for any arbitrary number of features, it may be difficult for datasets spanning a few hundred dimensions due to insufficient memory. For this reason, it may be essential to describe a method which extends the WMBS algorithm to high-dimensional datasets without significantly affecting its simplicity and accuracy.

In this paper, we employ the popular dimensionality reduction technique of Principal Component Analysis (PCA) to make the dataset more manageable for the WMBS algorithm. We argue that if the number of dimensions is comparable to the number of instances in the dataset, most of the features may be likely to be simple linear combinations of a subset of all the features. Since the WMBS algorithm inherently captures variances in the datasets being compared, it may be shown that the results obtained for a high-dimensional data are exactly the same as that obtained for a reduced dataset. This property forms the basis for the validity of our proposed method. We also compare this with other nonlinear approaches such as kernel PCA [12], probabilistic PCA [13], and t-Stochastic Neighborhood Embedding (t-SNE) [14], and show that a simple PCA-based reduction sufficiently captures similarity with the original dataset.

The remainder of this paper is organized as follows. In Section II, we briefly discuss the WMBS algorithm for multidimensional datasets, and provide a rigorous mathematical justification for its infeasibility for high-dimensional data. Section III summarizes the PCA method for dimensionality reduction, and also describes our proposed algorithm in some detail. In Section IV, we demonstrate our technique on two real UCI datasets comprising 281 and 500 features, respectively. We also argue why the original WMBS method may fail in practice for these and similar datasets, and discuss some stopping methods which may be used with the PCA module. We conclude with a discussion of possible improvements to the method in Section V.

## II. BACKGROUND

MFs may be categorized as crisp, T1 fuzzy [15], and T2 fuzzy [16], which may further be classified into interval T2 (IT2) [17] or general T2 (GT2), depending upon the shape assumed by the secondary MF. While T2 fuzzy MFs are better equipped for handling uncertainties of a higher order, they are also computationally more complex than T1 fuzzy MFs. IT2 fuzzy MFs have provided a convenient balance between this complexity and accuracy, and are therefore employed in several existing algorithms such as fuzzy c-means clustering [18], [19], [20], document categorization [21], and dynamic parameter adaptation for various optimization techniques [22],

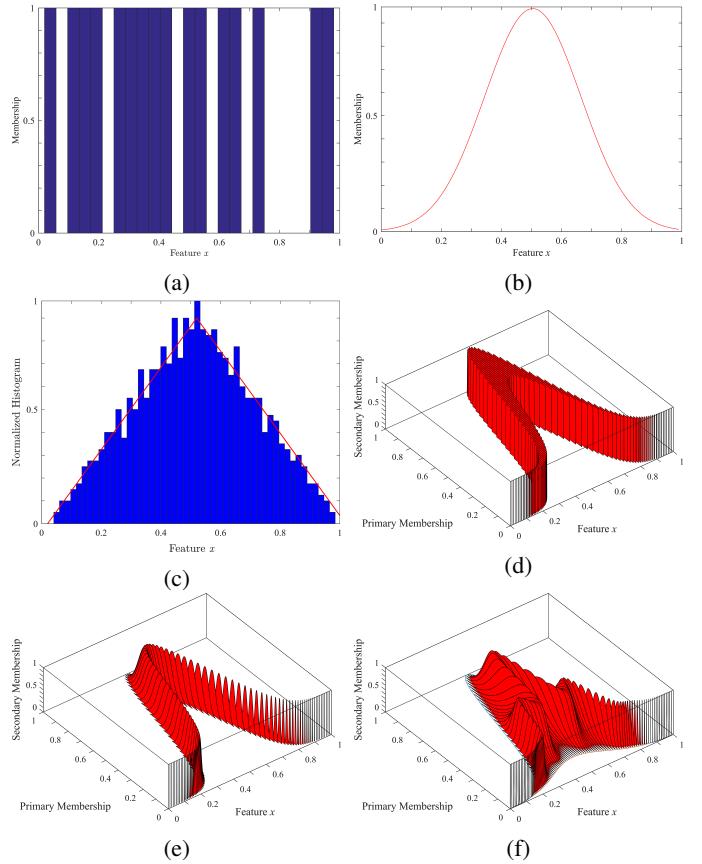


Fig. 1. Different types of membership functions: (a) crisp, (b) Gaussian T1 fuzzy, (c) triangular T1 fuzzy, (d) IT2 fuzzy, (e) symmetric Gaussian T2 fuzzy, and (f) asymmetric Gaussian “snaky” T2 fuzzy.

[23]. Fig. 1 shows various kinds of fuzzy MFs that may be generated for a given one-dimensional dataset.

Raj et al. [10] have proposed an iterative method consisting of three phases for suggesting the most appropriate fuzzy MF for a particular dataset and application. Fig. 2 outlines the method using a flow diagram, and the same is summarized as follows.

- 1) *Forward stage*: Various types of crisp and fuzzy MFs are generated from the “original” dataset.
- 2) *Reverse stage*: Data is generated back from each of the above MFs.
- 3) *Similarity analysis stage*: The  $p$ -value obtained from the hypothesis testing methods of Signed-Rank [24], [25] and Rank-Sum [26] tests, in conjunction with a decreasing bin size approach, is employed to determine the minimum bin size (i.e., maximum precision), for which the generated data may be considered similar to the original data.

The similarity analysis stage is logarithmic in complexity since the bin size used for computing the  $p$ -value is halved at each step until  $p$  becomes less than a predetermined threshold value (which is usually taken as 0.05 or 0.005 depending upon the variance in the dataset). The final bin size for which  $p$  is

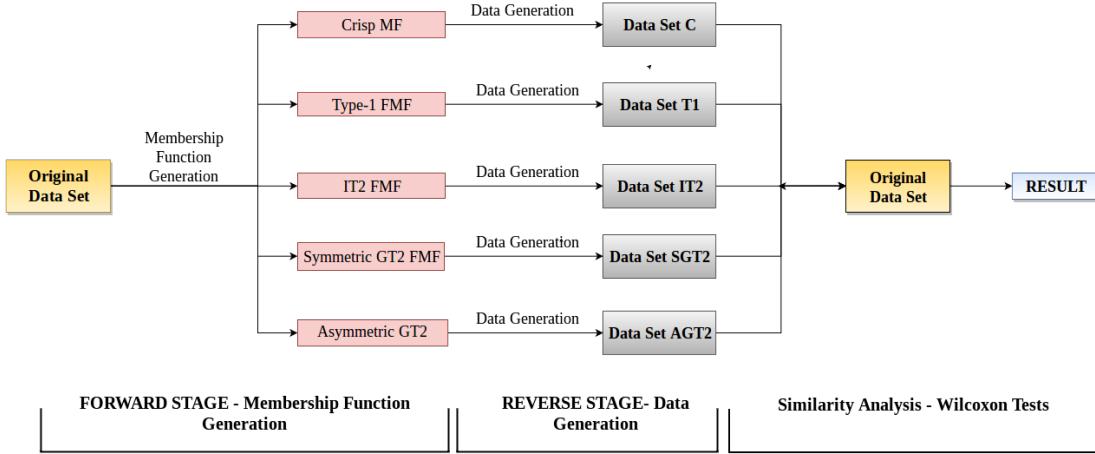


Fig. 2. Flow diagram for the Wilcoxon Minimal Bin Size algorithm for determining the most suitable fuzzy MF for a given dataset.

greater than the threshold is then returned, and compared with the bin sizes obtained for other fuzzy MFs. This method is described in Algorithm 1.

#### Algorithm 1 Iterative Algorithm for Determining Minimum Bin Size for Similarity

```

1: procedure WILCOXONMINIMALBINSIZE( $X_1, X_2$ )
2:    $n \leftarrow \text{size}(X_1)$ 
3:    $\maxValue \leftarrow \text{MAX}\{\text{MAX}(X_1), \text{MAX}(X_2)\}$ 
4:    $\text{binSize} \leftarrow \maxValue$ 
5:    $\text{dataIsSimilar} \leftarrow \text{true}$ 
6:   while  $\text{binSize} > 0$  AND  $\text{dataIsSimilar} == \text{true}$ 
    do
      7:      $m \leftarrow \maxValue/\text{binSize}$ 
      8:      $A, B \leftarrow \text{arrays of size } m \text{ initialized to 0}$ 
      9:     for  $0 \leq i \leq n$  do
       10:        $A[x_i/\text{binSize}] \leftarrow A[x_i/\text{binSize}] + 1$ 
       11:        $B[x_i/\text{binSize}] \leftarrow B[x_i/\text{binSize}] + 1$ 
       12:     end for
       13:     if  $\text{SIGNEDRANK}(A, B) == 0$  AND
           $\text{RANKSUM}(A, B) == 0$  then
       14:        $\text{binSize} \leftarrow \text{binSize}/2$ 
       15:     else
       16:        $\text{dataIsSimilar} \leftarrow \text{false}$ 
       17:     end if
       18:   end while
       19:   return  $\text{binSize}$ 
20: end procedure

```

We now analyze the performance of the algorithm on high-dimensional datasets, in order to justify the motivation behind this paper. Consider an  $m$ -dimensional dataset consisting of  $N$  samples. Without loss of generality, suppose the values of each dimension of all samples have been normalized between 0 and 1.

Two observations may be made at this point: (1) the method of similarity analysis is independent of the fuzzy MF being

used; and (2) the generation of a Gaussian T1 fuzzy MF and Gaussian T2 fuzzy MFs (symmetric and asymmetric) require the fitting of a Gaussian using the histogram approach, which further needs a complete view of all the dimensions in the dataset.

Suppose the WMBS algorithm terminates with  $b$  bins across each dimension, i.e., the dataset is divided into a total of  $b^m$  bins. From the above observations, it may be argued that in order to fit a Gaussian using a histogram approach, we require the values present in each of these  $b^m$  bins simultaneously. These values must therefore be present in memory at the same time, such that the memory used by any bin may not be used again for another bin.

In the case of a high dimensional dataset,  $m$  may be in the order of a few hundreds. Even with a conservative estimate, if  $m = 100$  and  $b = 2$ , the algorithm requires  $2^{100}$  (approximately  $10^{30}$ ) values to be stored simultaneously in memory. If each value requires a byte of storage space, the overall space complexity of the algorithm is  $10^{18}$  terabytes. To put this value in perspective, the total amount of data generated by humans to date is approximately  $10^9$  terabytes. Therefore, the WMBS algorithm in its original form may be difficult for high-dimensional datasets.

### III. PROPOSED METHOD

We employ the dimensionality reduction technique of Principal Component Analysis (PCA) [27] to make the WMBS algorithm applicable to high-dimensional datasets. PCA is mathematically defined [28] as “an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by some projection of the data lies on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on.”

There are several dimensionality reduction techniques other than PCA, such as kernel PCA and linear discriminant analysis (LDA), and ideally any of these methods may be employed for this purpose. However, the WMBS algorithm primarily works

with hyper-geometric Gaussians for the generation of most of the intermediate fuzzy MFs used in the forward and reverse stages, such as T1, SGT2, and “snaky” AGT2. It has been found that the PCA technique for dimensionality reduction works efficiently in the case of Gaussian data [29].

Implicitly, PCA assumes a constant multivariate normal distribution for variables of the data matrix  $\mathbf{X}$ . Data is centred with respect to a scaled sum of observations, representing the first moment estimator of a normal distribution. Loadings are then computed from eigen-decomposition of  $\mathbf{X}'\mathbf{X}$ , representing the second moment estimator of a normal distribution. No more moment estimators are used in calculations, concluding that a normal distribution is sufficient to describe the linear relationship between variables.

Each independent component of a multidimensional Gaussian is itself a Gaussian distribution, and PCA simply determines linear combinations of these Gaussians in such a way that most of the correlation between components is captured. Several methods have been proposed to determine the number of independent components that may be retained with PCA, some of which are as follows [30].

- 1) *Kaiser method*: The components with eigenvalues greater than 1 are retained.
- 2) *Scree test*: A scree plot displays the eigenvalues associated with a component or factor in descending order versus the number of the component or factor. The ideal pattern in a scree plot is a steep curve, followed by a bend and then a flat or horizontal line. The components in the steep curve before the flat line trend are retained.
- 3) *% of variation*: The components which cumulatively explain a certain percentage of variation are retained.
- 4) *Broken-stick method*: It assumes that if the total variance is divided randomly amongst the various components, then the expected distribution of the eigenvalues will follow a broken-stick distribution. Observed eigenvalues are considered interpretable if they exceed eigenvalues generated by the broken-stick model.

While the performance of any stopping method for PCA is greatly affected by the data distribution, the broken-stick method has been found to be the most promising since it provides a good combination of simplicity of calculation and accurate evaluation of dimensionality relative to other statistical approaches. For these reasons, we make use of this technique while applying PCA before the WMBS algorithm. We now describe this method in some detail.

**Broken stick model:** It was first presented by MacArthur [31] in the study of the structure of animal communities, specifically, bird species from various regions. Since each eigenvalue of a PCA represents a measure of each components’ variance, a component is retained if its associated eigenvalue is larger than the value given by the broken stick distribution.

Let us consider the closed interval  $J = [0, 1]$ . Now, suppose  $J$  is partitioned into  $n$  subintervals by randomly selecting  $n - 1$  points from a uniform distribution in this interval. If we arrange the subintervals according to the decreasing order of

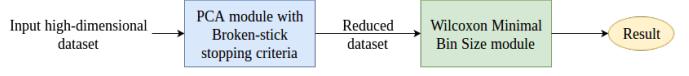


Fig. 3. Flow diagram for the extended WMBS algorithm with PCA module.

their lengths and denote by  $L_k$  the length of subinterval  $k$ , the expected value of  $L_k$  is given as

$$E(L_k) = \frac{1}{n} \sum_{j=k}^n \frac{1}{j}. \quad (1)$$

Although the broken stick model has been known to underestimate the dimension of the data, it is advantageous for our purposes since the space complexity of the WMBS algorithm increases exponentially with the number of features.

The extended WMBS algorithm is then summarized as a two-stage process as follows. The method is also represented pictorially in the flow diagram shown in Fig. 3.

*Stage 1*: The high-dimensional dataset is passed as input to a PCA module with a broken-stick stopping criterion. Since most real distributions are not random in the sense that a large fraction of the variance is captured by a small number of components, it may be argued that the PCA module generates a dataset which has significantly less number of features than the original data.

*Stage 2*: The reduced dataset is given to a WMBS module which determines the most appropriate fuzzy MF to be used with the given dataset through the forward, reverse, and similarity analysis stages.

#### IV. EXPERIMENTAL RESULTS

We now validate our proposed method on two high-dimensional datasets, namely BlogFeedback [32] and Madelon [33], which have been obtained from the UCI machine learning repository [34]. The datasets are described as follows.

- 1) *BlogFeedback*: Instances in this dataset contain 281 features extracted from 60021 blog posts. The task is to predict how many comments the post will receive.
- 2) *Madelon*: It is an artificial dataset which was part of the NIPS 2003 feature selection challenge. This is a two-class classification problem with continuous input variables, consisting of 4400 instances and 500 features.

When these datasets are passed through the PCA module, the obtained eigenvalues for the independent components, and their corresponding broken-stick values are represented in Figs. 4(a) and 4(b), respectively. Although the datasets consist of 281 and 500 features, respectively, we represent only 20 dimensions for each to compare with the values obtained from the broken stick method, for sake of clarity. Further, the eigenvalues in Fig. 4(a) are represented on a logarithmic scale since the value of the first 2 components are exponentially high compared to others.

From the figures, it may be observed that the broken stick stopping criteria suggests 2 and 3 independent components for the BlogFeedback and Madelon datasets, respectively. Figs.

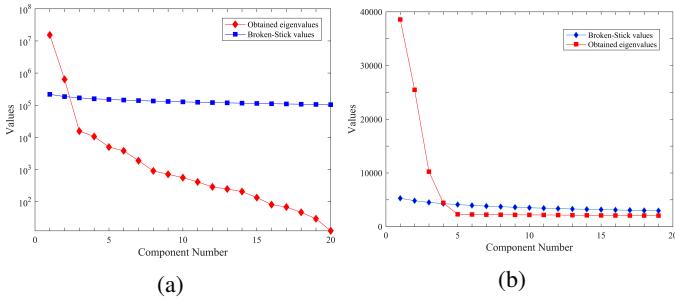


Fig. 4. Obtained eigenvalues and broken-stick values for first 20 independent components for (a) BlogFeedback and (b) Madelon datasets.

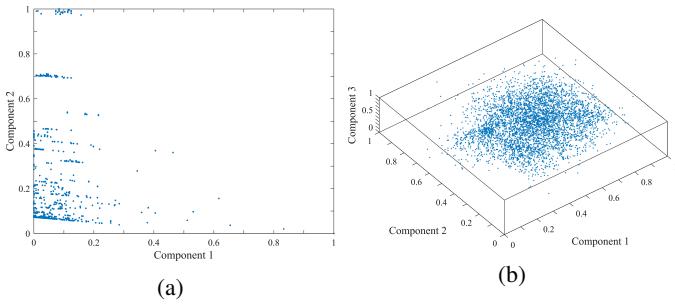


Fig. 5. Scatter plots of reduced datasets obtained by applying the PCA module with broken-stick stopping criteria on (a) BlogFeedback and (b) Madelon datasets.

5(a) and 5(b) show the scatter plots for the corresponding reduced data obtained after the PCA module.

We apply the WMBS algorithm on the reduced BlogFeedback dataset and the results for various bin sizes are shown in Table I. The  $p$ -value denotes the probability of observing a test statistic, as or more extreme than the observed value under the null hypothesis, and statistic  $h$  denotes the result of the hypothesis test. If  $h = 1$ , it indicates the rejection of the null hypothesis at the  $100\alpha\%$  significance level. It may be noted that a T1 fuzzy MF is best able to approximate the reduced dataset for  $n=0.2$ .

In Table II, we compare a PCA-based reduction with other dimensionality reduction techniques to further the argument in favor of this simple method. The data is reduced to both 3 and 4 dimensions for sake of comparison, and it may easily be observed that the similarity in case of PCA reduction is comparable, if not higher, even for small bin sizes, than complex nonlinear methods such as kernel PCA, probabilistic PCA, and t-SNE.

If the original WMBS algorithm was employed for determining the fuzzy MF generating maximum similarity, the task may be challenging due to memory requirements for the given size of data. Our proposed method is, therefore, efficient and effective for datasets of similar dimensionalities as these. Further, an ensemble of PCA modules with various stopping criteria may be used instead of a single module with the broken-stick model.

TABLE I.  
WILCOXON TEST STATISTICS FOR BLOG FEEDBACK DATA  
VS DATA SETS GENERATED FROM T1, IT2, AND SGT2  
FUZZY MFs.

Bin Size ( $n$ )	Test Name	Test Statistics	DATA vs T1 data	DATA vs IT2 data	DATA vs SGT2 data
0.5	WSR	$p$	0.3086	0.3086	0.3613
		$h$	0	0	0
	WRS	$p$	0.1516	0.1400	0.1638
		$h$	0	0	0
0.25	WSR	$p$	0.7801	0.4354	0.5298
		$h$	0	0	0
	WRS	$p$	0.5914	0.7896	0.6854
		$h$	0	0	0
0.2	WSR	$p$	0.8048	0.6970	0.5926
		$h$	0	0	0
	WRS	$p$	0.2596	0.0361	0.0273
		$h$	0	1	1
0.1	WSR	$p$	7.38e-14	-	-
		$h$	1	-	-
	WRS	$p$	1.15e-09	-	-
		$h$	1	-	-

The various components within this ensemble may then be weighted according to their performance in determining the number of independent features.

## V. CONCLUSION AND FUTURE WORK

In this paper, we proposed and validated a PCA-based method to further extend the Wilcoxon's Minimal Bin Size algorithm to datasets consisting of several hundred features. While dimensionality reduction is a popular technique for high-dimensional datasets, its applicability to hypothesis testing has not been rigorously proven, which may raise questions about the robustness of the PCA-WMBS method. However, it may be argued that if the original data is in the form of an ellipsoid, PCA may be likely to be effective since it preserves the Gaussian nature of the distribution through linear transformations. This was evident from experimentation on the Madelon dataset. Furthermore, comparison with other complex methods such as kernel PCA, probabilistic PCA, and t-SNE showed the effectiveness of the simple PCA method for simple datasets.

If the high-dimensional dataset is inherently more random (such as the BlogFeedback data), an ensemble of several dimensionality reduction techniques may be applied to obtain a better low-dimensional representation of the data than what is obtained from PCA. Analysis of the results obtained from the different module may then be used to assign weights to the various components in the ensemble. It may also provide an accurate insight into the data distribution itself.

## ACKNOWLEDGMENT

This work was supported by the Technology Innovation Program of the Korea Institute for Advancement of Technology(KTA) granted financial resource from the Ministry of Trade, Industry & Energy, Republic of Korea (No. 2015-122).

TABLE II.  
WILCOXON TEST STATISTICS FOR MADELON DATA VS DATA SETS GENERATED FROM T1, IT2, AND SGT2 FUZZY MFs.

DR Method	3 Dimensions				4 Dimensions			
	Type	Bin Size	p-WSR	p-WRS	Type	Bin Size	p-WSR	p-WRS
PCA	T1	0.1	0.9418	0.2886	<b>T1</b>	<b>0.5</b>	<b>0.9588</b>	<b>0.9358</b>
	<b>IT2</b>	<b>0.05</b>	<b>0.2562</b>	<b>0.8834</b>	IT2	0.5	0.8767	0.8850
	SGT2	0.05	0.1801	0.7375	SGT2	0.5	0.8767	0.8951
k-PCA	T1	0.1	0.8749	0.2377	<b>T1</b>	<b>0.5</b>	<b>0.3314</b>	<b>0.2702</b>
	<b>IT2</b>	<b>0.05</b>	<b>0.6112</b>	<b>0.7714</b>	IT2	0.5	0.0645	0.2222
	SGT2	0.05	0.1486	0.3162	SGT2	0.5	0.2695	0.1836
p-PCA	<b>T1</b>	<b>0.1</b>	<b>0.7672</b>	<b>0.2073</b>	<b>T1</b>	<b>0.5</b>	<b>1</b>	<b>0.8065</b>
	IT2	0.2	0.0676	0.5904	IT2	0.5	0.8361	0.6376
	SGT2	0.1	0.5488	0.1643	SGT2	0.5	1	0.7774
t-SNE	T1	0.2	0.1634	0.7829				
	<b>IT2</b>	<b>0.2</b>	<b>0.3062</b>	0.7372				
	<b>SGT2</b>	<b>0.2</b>	0.1563	<b>0.7842</b>				

## REFERENCES

- [1] J. Ahn, Y. Yuan, G. Parmigiani, M. B. Suraokar, L. Diao, I. I. Wistuba, and W. Wang, "Demix: deconvolution for mixed cancer transcriptomes using raw measured data," *Bioinformatics*, vol. 29, no. 15, pp. 1865–1871, 2013.
- [2] M. Newton, A. Noueiry, D. Sarkar, and P. Ahlquist, "Detecting differential gene expression with a semiparametric hierarchical mixture method," *Biostatistics*, vol. 5, no. 2, pp. 155–176, 2004.
- [3] G. Smyth *et al.*, "Linear models and empirical bayes methods for assessing differential expression in microarray experiments," *Statistical Applications in Genetics and Molecular Biology*, vol. 3, no. 1, pp. 1–25, 2004.
- [4] J. Leek and J. Storey, "Capturing heterogeneity in gene expression studies by surrogate variable analysis," *PLoS Genetics*, vol. 3, no. 9, pp. 1724–1735, 2007.
- [5] H. Hagras, "A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 4, pp. 524–539, 2004.
- [6] N. Karnik and J. Mendel, "Applications of type-2 fuzzy logic systems to forecasting of time-series," *Info. Sci.*, vol. 120, no. 1, pp. 89–111, 1999.
- [7] R. John, P. Innocent, and M. Barnes, "Neuro-fuzzy clustering of radiographic tibia image data using type 2 fuzzy sets," *Info. Sci.*, vol. 125, no. 1, pp. 65–82, 2000.
- [8] D. Wu and W. Tan, "Type-2 FLS modeling capability analysis," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2005, pp. 242–247.
- [9] J. Mendel, "A quantitative comparison of interval type-2 and type-1 fuzzy logic systems: First results," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2010, pp. 1–8.
- [10] D. Raj, A. Gupta, B. Garg, K. Tanna, and F. C.-H. Rhee, "Analysis of data generated from multidimensional type-1 and type-2 fuzzy membership functions," *IEEE Trans. Fuzzy Syst.*, DOI:10.1109/TFUZZ.2017.2688342, 2017.
- [11] D. Raj, K. Tanna, B. Garg, and F. C.-H. Rhee, "Visual analysis and representations of type-2 fuzzy membership functions," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2016, pp. 550–554.
- [12] B. Schölkopf, A. Smola, and K. Müller, "Kernel principal component analysis," in *International Conference on Artificial Neural Networks*. Springer, 1997, pp. 583–588.
- [13] M. Tipping and C. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.
- [14] L. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.
- [15] J. Mendel, *Uncertain rule-based fuzzy logic systems: introduction and new directions*. Prentice Hall PTR Upper Saddle River, 2001.
- [16] N. Karnik and J. Mendel, "Introduction to type-2 fuzzy logic systems," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, vol. 2. IEEE, 1998, pp. 915–920.
- [17] Q. Liang and J. Mendel, "Interval type-2 fuzzy logic systems: theory and design," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5, pp. 535–550, 2000.
- [18] C. Hwang and F. C.-H. Rhee, "Uncertain fuzzy clustering: Interval type-2 fuzzy approach to *c*-means," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 1, pp. 107–120, 2007.
- [19] B. Choi and F. C.-H. Rhee, "Interval type-2 fuzzy membership function generation methods for pattern recognition," *Information Sciences*, vol. 179, no. 13, pp. 2102–2122, 2009.
- [20] F. C.-H. Rhee, "Uncertain fuzzy clustering: Insights and recommendations," *IEEE Computational Intelligence Magazine*, vol. 2, no. 1, pp. 44–56, 2007.
- [21] V. Pham, L. Ngo, and W. Pedrycz, "Interval-valued fuzzy set approach to fuzzy co-clustering for data classification," *Knowledge-Based Systems*, 2016.
- [22] C. Peraza, F. Valdez, and O. Castillo, "Interval type-2 fuzzy logic for dynamic parameter adaptation in the harmony search algorithm," in *Proc. IEEE Int. Conf. Intelligent Syst.*, 2016, pp. 106–112.
- [23] F. Olivas, F. Valdez, O. Castillo, and P. Melin, "Dynamic parameter adaptation in particle swarm optimization using interval type-2 fuzzy logic," *Soft Computing*, vol. 20, no. 3, pp. 1057–1070, 2016.
- [24] R. Randles, "Wilcoxon signed rank test," *Encyclopedia of Statistical Sci.*, 1988.
- [25] B. Rosner, R. Glynn, and M. Lee, "The wilcoxon signed rank test for paired comparisons of clustered data," *Biometrics*, vol. 62, no. 1, pp. 185–192, 2006.
- [26] W. Haynes, "Wilcoxon rank sum test," in *Encyclopedia of Systems Bio*. Springer, 2013, pp. 2354–2355.
- [27] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent lab. syst.*, vol. 2, no. 1-3, pp. 37–52, 1987.
- [28] I. Jolliffe, *Principal Component Analysis*. Wiley Online Library, 2002.
- [29] R. Rummel, *Applied factor analysis*. Northwestern University Press, 1988.
- [30] D. Jackson, "Stopping rules in principal components analysis: a comparison of heuristical and statistical approaches," *Ecology*, vol. 74, no. 8, pp. 2204–2214, 1993.
- [31] R. MacArthur, "On the relative abundance of bird species," *Proc. National Academy of Sci.*, vol. 43, no. 3, pp. 293–295, 1957.
- [32] K. Buza, "Feedback prediction for blogs," in *Data analysis, machine learning and knowledge discovery*. Springer, 2014, pp. 145–152.
- [33] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror, "Result analysis of the nips 2003 feature selection challenge," in *NIPS*, vol. 4, 2004, pp. 545–552.
- [34] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: <http://archive.ics.uci.edu/ml>