Influence of Noisy and High-Dimensional Data on Semi-Supervised Learning

Tianya Hou and Hyunjung Shin*

***Abstract*—Semi-supervised learning (SSL) has become a popular tool in machine learning. In SSL, label prediction for an unlabeled data point is determined by the similarities from its adjacent data points, thus how to make the similarity matrix is a critical factor determining the performance of SSL. When the data is noisy and has high dimensionality, however, the similarity matrix is no longer reliable and consequently, it is hard to expect reasonable performance of SSL. To overcome the difficulties, we propose to make the similarity matrix with the extracted features either from principal component analysis (PCA) or nonlinear principal component analysis (NLPCA). The method was validated on one artificial- and five real-world-problems. Thanks to the newly made similarity matrix based on the extracted features, the performance of SSL becomes robust against noise features and high dimensionality**.

I. INTRODUCTION

RECENTLY, semi-supervised learning (SSL) has become a popular tool in machine learning, particularly in domains such as text classification and bioinformatics [5]-[6]. Given sets of labeled and unlabeled data points, the task of predicting the missing labels can be aided by the information from unlabeled data points, for example, by using information about the *manifold structure* of the data in input space. Many state-of-the art methods implement a SSL approach in that they incorporate information from unlabeled data points into the learning paradigm [1]-[4]. Despite their many variations, one thing common to most algorithms is the use of a matrix of values representing the pairwise relationships between data points. The matrix is denoted as the "similarity matrix" and plays a critical role in determining the performance of SSL. The label prediction of an unlabeled data point is made through the propagation of the labels of its adjacent data points, and the strength of influence of each is proportional to the similarity between them. However, the similarity matrix can be easily affected by the noise features and high dimensionality of the raw dataset [7]. If irrelevant or highly correlated features are included in calculating the similarity, and even worse if the data is high- dimensional, the similarity matrix no more well reflects the points' influence on each due to either "noisy influence" or the "curse of dimensionality." Consequently, it can probably deteriorate the generalization ability of SSL.

To circumvent this, a preventative against noisy influence and the curse of dimensionality becomes necessary before making the similarity matrix. One method is to use the feature extraction (dimensionality reduction) techniques in the preprocessing step. Feature extraction refers to the process of finding a mapping that reduces the dimensionality and of removing the noise effect from the dataset. With the newly extracted features, we can exclude the influence of redundant or noisy features from the similarity matrix with little or no loss of information [8], [9], and therefore, we can expect a better performance of SSL. There are various kinds of the method for feature extraction. As a linear method, principal component analysis (PCA) is the representative. By calculating the eigenvectors of the covariance matrix of the original data, PCA linearly transforms a high-dimensional vector of the input features into a low-dimensional one whose components (extracted features) are uncorrelated [10], [11]. On the contrary, there are several types of implementation of nonlinear PCA (NLPCA) [12]. Autoassociative neural network (AANN) is one of the well-known nonlinear transformation methods. In AANN, the network is trained to perform the identity mapping where the values of input features are approximated at the output layer, and the nonlinear principal components can be obtained from the hidden nodes in the bottleneck layer [13]-[16].

The primary purpose of this paper is to diagnose how robust an SSL algorithm against the noisy and high-dimensional input features, and therefrom suggest to use PCA or NLPCA as a preventative. Through feature extraction, the data points in the higher input space are projected onto a lower dimensional space of the new features extracted either from PCA or NLPCA. In the experiment, the performances of SSL with the original features and with the extracted features are compared. Superiority of PCA to NLPCA (or vice versa) depends upon linearity (or nonlinearity) of the intrinsic features of the underlying problem, but in practice it is difficult to know *a priori* that the given problem is linear or nonlinear. Therefore, the performance comparison between the two extraction methods will not of great interest in this paper.

The rest of this paper is organized as follows. In section 2, the theory of SSL is presented. In section 3, PCA and AANN, the feature extraction methods, are described, respectively. Section 4 provides the experimental results, followed by the conclusions in the last section.

II. SEMI-SUPERVISED LEARNING

In (graph based) semi-supervised learning algorithm, a data point $\boldsymbol{x_i} \in R^m$ *(i = 1, ..., n)* is represented as a node *i* in a graph,

This work was supported in part by Post Brain Korea 21 and the research grant form Korean Government (MOEHRD, Basic Research Promotion Fund, KRF-2008-531-D00032).

Tianya Hou (houtianya@ ajou.ac.kr) is in the master course of Industrial & Information Systems Engineering, Ajou University, Suwon, 443-749 Korea

* Corresponding author: Hyunjung Shin (shin@ajou.ac.kr) is a professor of the department of Industrial & Information Systems Engineering, Ajou University, Suwon, 443-749, Korea

and the relationship between data points is represented by an edge where the connection strength from each node *j* to each other node *i* is encoded as w_{ij} of a weight matrix *W*. A weight w_{ij} can take a binary value (*0* or *1*) in the simplest case. Often, a Gaussian function of Euclidean distance between points with length scale σ is used to specify connection strength:

$$w_{ij} = \begin{cases} exp\left(-\frac{(x_i - x_j)^T (x_i - x_j)}{\sigma^2}\right) & if\ i \sim j \\ 0 & otherwise \end{cases}, \quad (1)$$

The $i \sim j$ stands for node *i* and *j* having an edge between them which can be established either by k nearest neighbors or by Euclidean distance within a certain radius *r*, $\|x_i - x_j\|^2 < r$. The labeled nodes have labels $y_l \in \{-1,1\}\ (l = 1, \dots, L)$, while the unlabeled nodes have zeros $y_u = 0$ *(u=L+1,...,L+U)*. The algorithm will output an n-dimensional real-valued vector $f = [f_l^T\ f_u^T]^T = (f_1, \dots, f_L, f_{L+1}, \dots, f_{L+U})^T$ which can be thresholded to make label predictions on $f_{L+1}, \dots, f_{L+U}$ after learning. It is assumed that (a) f_i should be close to the given label y_i in labeled nodes and (b) overall, f_i should not be too different from its adjacent nodes f_js (i∼j). One can obtain f by minimizing the following quadratic functional:

$$Min_f\ \ (f - y)^T (f - y) + \mu f^T L f\ , \quad (2)$$

where $y = (y_1, \dots, y_l, 0, \dots, 0)^T$, and the matrix $\boldsymbol{L}$, called the graph Laplacian matrix, is defined as $L = D - W$, where $D = \mathrm{diag}(d_i)$, $d_i = \sum_j \omega_{ij}$. The first term corresponds to the loss function in terms of condition (a), and the second term represents the smoothness of the predicted outputs in terms of condition (b). The parameter μ trades off loss versus smoothness. The solution of this problem is obtained as

$$f = (I + \mu L)^{-1} y\ , \quad (3)$$

where I is the identity matrix.

III. FEATURE EXTRACTION

A. Principal Component Analysis (PCA)

PCA can be used for dimensionality reduction in a data set by extracting important hidden features that contribute most to its variance. Technically, PCA attempts to find orthonormal axes which maximally decorrelate the original features of the data. Given the input data points $x_i \in R^m$ $(i = 1, \dots, n$ and $\sum_{i=1}^n x_i = 1,$ usually $m < n)$, PCA linearly transforms each data point x_i into a new one s_i by

$$\underbrace{s_j}_{m \times 1} = \underbrace{U^T}_{m \times m}\ \underbrace{x_j}_{m \times 1}\ , \quad i = 1, \dots, n\ , \quad (4)$$

where U is the $m \times m$ orthogonal matrix whose kth column u_k is the kth eigenvector of the covariance matrix $C = \frac{1}{n}\sum_{i=1}^n x_i x_i^T$. The matrix U can be obtained by solving the eigenvalue problem on C,

$$\lambda_k u_k = C u_k, \qquad k = 1, \dots, m\ , \quad (5)$$

where λ_k is an eigenvalue of C and u_k is the corresponding eigenvector. The magnitude of an eigenvalue stands for the proportion of variance that can be explained by the corresponding eigenvector. Therefore, when taking the first p eigenvectors $\tilde{U}^T = \{u_1, u_2, \dots, u_p\}$ referring to the descending order of eigenvalues, $\lambda_1 > \lambda_2 > \cdots > \lambda_p > \cdots > \lambda_m$, we can find "lower" dimensional orthonormal space (m→p) yet still retaining most important aspects of the data. A projected data point onto the lower dimensional space, $\tilde{s}_i$, is calculated as the orthogonal transformations of x_i,

$$\underbrace{\tilde{s}_j}_{p \times 1} = \underbrace{\tilde{U}^T}_{p \times m}\ \underbrace{x_j}_{m \times 1}\ , \quad i = 1, \dots, n\ , \quad (6)$$

PCA is a well-established dimensionality reduction method. However, its applicability is limited by the assumptions on linearity that the data set to be "linear" combinations of certain features. Therefore, if the data set shows non-linear relationship among features, there is no guarantee that the extracted features by PCA will contain important features.

B. Nonlinear Principal Component Analysis: Autoassociative Neural Network (AANN)

Another approach to dimensionality reduction is through the use of an autoassociative neural network (AANN), a special kind of feed-forward neural networks [17]. AANN finds and eliminates nonlinear correlations in the data. Analogous to principal component analysis, it can be used to reduce the dimensionality of data by removing redundant features. General structure of AANN is shown in Fig. 1. It consists of an input layer, an output layer, and multiple hidden layers. Both the number of input nodes and that of output nodes are equally set to *m*. Among the hidden layers, the mapping layer models the mapping function (F_1) and the demapping layer models the demapping function (F_2). The number of nodes in a particular hidden layer (*p*), so called "bottleneck layer", is set to be less than the number of nodes in the input/output layer (p<*m*).

In autoassociative mapping, the target data are set to be identical to the input data. This "identity mapping" creates a global reduction of the data dimensionality while the input data go through the bottleneck layer before appearing at the output layer. Let *F* denote the autoassociative mapping learnt by the network. If $\{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n\}$ is the set of output data produced by the AANN when the input data set $\{x_1, x_2, \dots, x_n\}$ is given, the *F* can be found while minimizing the mean square error *E*,

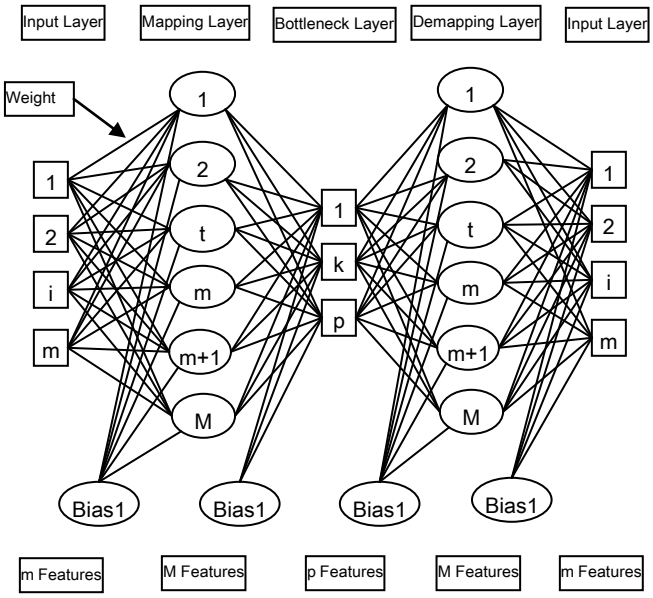

Fig.1 Architecture of AANN

$$E = \sum_{i=1}^{n}(\boldsymbol{x_i} - \tilde{\boldsymbol{x}}_i)^T(\boldsymbol{x_i} - \tilde{\boldsymbol{x}}_i) = \sum_{i=1}^{n}(\boldsymbol{x_i} - F(\boldsymbol{x_i}))^T(\boldsymbol{x_i} - F(\boldsymbol{x_i})). \quad (7)$$

The mapping function F can be separated into F_1 and F_2, so that $F(.) = F_2(F_1(.))$, where F_1 is the transformation in the network from the input layer upto the dimension compressing hidden layer (the bottleneck layer), and F_2 is the transformation from the bottleneck layer upto the output layer. To summarize, the data is first compressed to lower dimensionality and then reconstructed. The mapping from the input layer to the bottleneck layer can be regarded as "nonlinear" projection onto the lower dimensional space (m→p), and each node in the bottleneck can be considered as an extracted feature retaining significant information of the data. New data $\tilde{s}_i$ are then calculated as

$$\underbrace{\tilde{s}_l}_{p\times 1} = F_1\underbrace{(x_l)}_{m\times 1}, \quad (8)$$

In theory, AANN can extract good features from any type of nonlinear relationship occurring in the data if the architecture is well designed. However, there is no definitive method for deciding *a priori* the number of nodes in the bottleneck layer.

IV. EXPERIMENT RESULTS

We applied the proposed method to various kinds of problems: an artificial problem and five real-world problems. The original dimensionality of each dataset was reduced by using PCA and NLPCA respectively, and the similarity matrices from the original features and the two types of new extracted features were calculated. Hereafter, we denote those matrices as $\boldsymbol{W}_{\text{original}}$, $\boldsymbol{W}_{\text{PCA}}$ and $\boldsymbol{W}_{\text{NLPCA}}$. For performance comparison of SSL among the three different similarity matrices, the area under the ROC curve (AUC) were measured. This experimental setting will show how influential the similarity matrix is to the performance of SSL, particularly when the original data is noisy and the dimensionality is high.

A. *Artificial problem*

The proposed method was evaluated on Two-moon data as shown in Fig. 2. A total of *500* two-dimensional data points were generated from two underlying classes and each class has *250* data points. We added different numbers of distractor features, called 'noises', which have nothing to do with classification accuracy. The number of the noise features varies depending on the degree of noise, 13 and 52, respectively, as shown in Fig. 3.

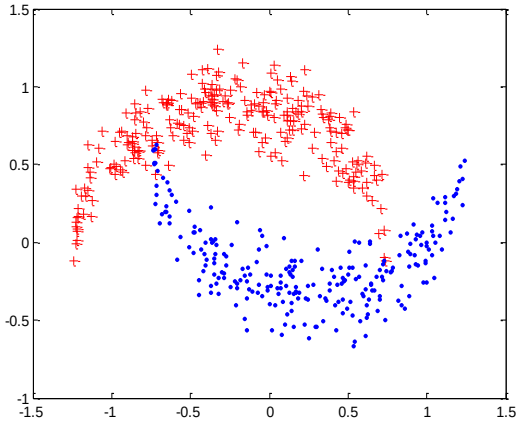

Fig. 2 Two-moon data

According to the experimental setting, several types of similarity matrix were calculated from the original two features (two-dimension), the 13 noise-added features (15-dimension), the 52 noise-added features (54-dimension), the two sets of the PCA extracted two features from 13 (or 52) noise-added feature set (two-dimension) and the two sets of NLPCA extracted two features from 13 (or 52) noise-added feature set (two-dimension), respectively. Then the AUCs of seven similarity matrices were measured under various combinations of parameters $\{k, \mu, N\}\epsilon\{3,5,10\} \times \{0.1,1,10\} \times \{2,5,10,20\}$ where k is the number of k-nearest neighbors in Eq.(1), μ is the loss-smoothness tradeoff parameter in Eq.(2) and N is the percentage (%) of the labeled data points in the data set.

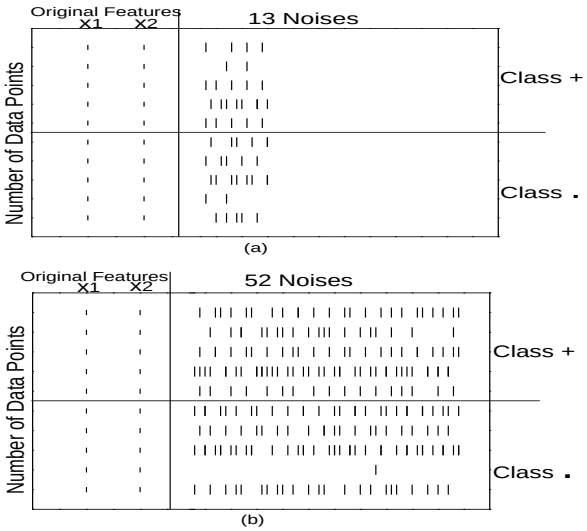

Fig. 3 Experimental setting with two different degrees of noise added to the two original features, (a) 13 noise features and (b) 52 noise features, respectively.

The results are shown in Fig. 4. For simplicity, the similarity matrix of the original two features is denoted as '$\boldsymbol{W}_{original}$', that of the noise-added features as '$\boldsymbol{W}_{w/noise}$' and those of the extracted features through PCA and NLPCA as '$\boldsymbol{W}_{PCA}$' and '$\boldsymbol{W}_{NLPCA}$', respectively. Fig. 4(a) shows the results for the case of the 13 noise-added features. The average AUC of $\boldsymbol{W}_{original}$ is 0.952, while the average AUC of $\boldsymbol{W}_{w/noise}$ is 0.734. And so the AUC decreases 22.0% after noise feature addition. This implies that the existence of noise features degrades the original performance of SSL.

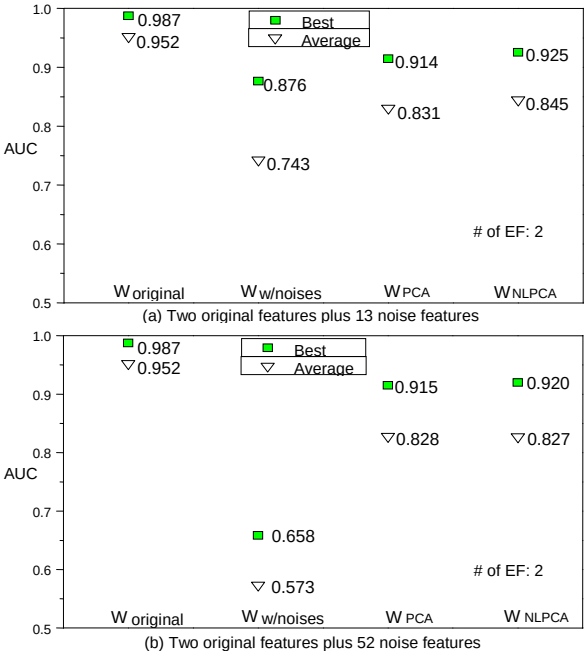

Fig. 4 The comparison of AUC for different similarity matrices. The EF stands for the extracted features through PCA or NLPCA. The square indicates the best AUC after repetition of experiments over every combination of parameters, and the reverse triangle indicates the average AUC.

The best AUCs of $\boldsymbol{W}_{w/noise}$ is 0.876, while those of W_{PCA} and W_{NLPCA} are 0.914 and 0.925, respectively. Both methods improve the accuracy. In Fig. 4(b), when the number of the added noise features is 52, the avg. AUC for $\boldsymbol{W}_{w/noise}$ becomes considerably lower than that of $\boldsymbol{W}_{original}$ (0.952 vs. 0.573). The amount of the decrease is about 39.8% of the original performance. However, no matter which feature extraction method is used, the large amount of original performance is regained: 0.828 for W_{PCA} and 0.827 for W_{NLPCA}. The best AUC of $\boldsymbol{W}_{w/noise}$ is 0.658 while those of W_{PCA} and W_{NLPCA} are 0.915 and 0.920, respectively. Through feature extraction, the AUC can be increased by 39.6% in the best case.

Comparing the two feature extraction methods, NLPCA performed slightly better than PCA. The reason can be explained by that the underlying features of Two moon problem are nonlinear. And NLPCA is better at discovering nonlinear features than PCA which is based on linear transformation. However, it is hard to know in advance whether the problem in hand is linear or nonlinear.

When comparing the two sets of experiments, the 13 noise added features and the 52 noise-added features, we can conclude that the more noise features incur the more serious degradation in performance of SSL. To recover the loss, feature extraction can be employed. Either PCA or NLPCA recovers most of the original performance. In our experiment, we used only "two" extracted features (EF), but we can expect better performance with more extracted features. And, it is interesting to see that the performance of feature extraction method is not sensitive to the changes of degree of noise from 13 to 52: 0.831 to 0.828 for PCA and 0.845 to 0.827 for NLPCA. This means the two feature extraction methods successfully found the intrinsic dimensions of the problem (two dimensions in our problem) in both experimental settings. Therefore, we can expect more stabilized performance of SSL through feature extraction regardless of the degrees of noise or high dimensionality.

B. Real-world problems

a. Data

Five real-world data sets were used for benchmarking. Table 1 summarizes the data sets from diverse fields: Pima Indians Diabetes, SPECTF and WDBC are available at [18] and Digit1 and USPS are available at [19].

TABLE I
FIVE REAL-WORLD DATA SETS

Data set	Classes	Original features	Data Points	New Features
Pima Indians Diabetes	2	8	768	3
SPECTF	2	44	267	4
WDBC	2	32	569	7
Digit1	2	241	1500	20
USPS	2	241	1500	20

In order to increase readability for the results, the AUCs are shown at a fixed set of the values of hyperparameters, $\{k, \mu, N\}$ at (*10, 1, 10%*).

b. The number of extracted features

It is difficult to determine the number of extracted features when the intrinsic dimension is unknown as in usual real-world problems. One of the rule-of-thumbs for PCA is to draw a scree plot and find the elbow point to determine the appropriate number of features to be extracted. In a scree plot, the proportion of all eigenvalues is drawn in their decreasing order. The plot looks like the side of a mountain, and "scree" refers to the debris falling from a mountain and lying at its base. So it proposes to determine the number of extracted features at the point the mountain ends and debris begins [20]. For AANN, however, it is hardly known how to determine the number of hidden nodes in the bottleneck layer. In our experiments, the elbow points for PCA were also used to determine the number of hidden nodes for NLPCA.

Fig. 5 shows the resulting scree plots for the five data sets. As shown in the plots, the elbows for Pima, SPECTF, WDBC, Digit1, and USPS were found at 3, 4, 7, 20, and 20 respectively, and they became the number of the features to be extracted for both PCA and NLPCA.

c. Accuracy

The AUC results for the five real-world data sets are shown in Fig. 6. For Pima data set, both W_{PCA} and W_{NLPCA} achieved a reasonable accuracy in AUC. Similar results can be found at WDBC, Digit1 and USPS. For SPECTF data set, there is a pronouncing improvement in AUC: 0.528 for $W_{original}$ jumped upto 0.68 and 0.71 for W_{PCA} and W_{NLPCA}, respectively. We may have a conjecture that SPECTF data set contains a lot of redundant or noise features, and so the PCA or NLPCA properly works to extract relevant features among them. Across the five data sets, there is a slight competition between W_{PCA} and W_{NLPCA}. For instance, W_{PCA} outperforms W_{NLPCA} in SPECTF (0.68 vs. 0.71). On the contrary, W_{NLPCA} outperforms W_{PCA} in USPS (0.95 vs.0.92). Superiority of one method to the other seems to be dependent upon linearity (or nonlinearity) of the intrinsic features of the underlying problem. A more important fact we found through the experimental results is that W_{PCA} or W_{NLPCA} enables us to obtain a similar or better accuracy with much less number of features. The number of the extracted features ranges from about 8% to 38% of the original number of features: 8% in either Digit1 or USPS and 38% in Pima. This will be studied further in the next section. Table I shows the details. The Wilcoxon signed-ranks test were used to compare W_{PCA} (or W_{NLPCA}) and $W_{original}$ [21]. A smaller *p*-value stands for W_{PCA} (or W_{NLPCA}) outperforms $W_{original}$ with a greater statistical significance.

d. Efficacy of a feature

The results in Fig.5 and table I showed that a similar or better accuracy can be obtained with much less number of features through W_{PCA} or W_{NLPCA}. In order to study efficacy of a feature in length, we defined "CF" as follows,

$$CF = \frac{AUC}{number\ \ of\ features}, \qquad (9)$$

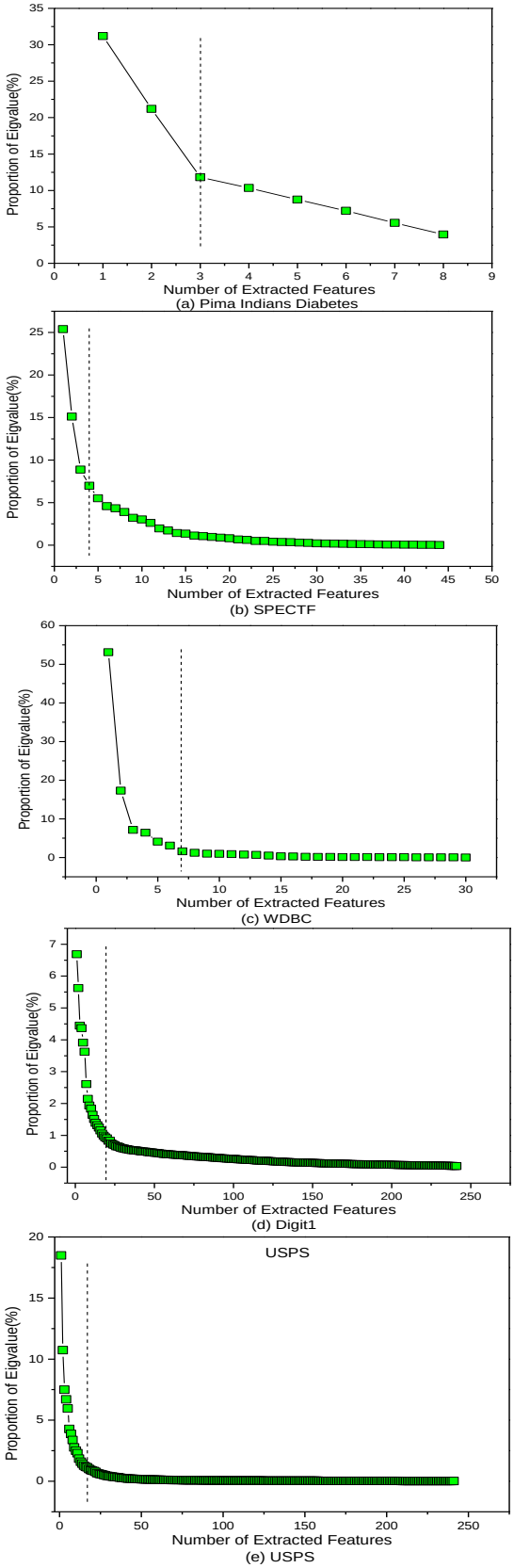

Fig. 5 Scree plots for five data sets: the dotted line indicates the number of the features to be extracted, 3 for Pima Indian Diabetes, 4 for SPECTF, 7 for WDBC, 20 for Digit1, and 20 for USPS.

where it quantifies the amount of contribution of a feature to overall accuracy. A larger value of CF means a higher efficacy of a feature.

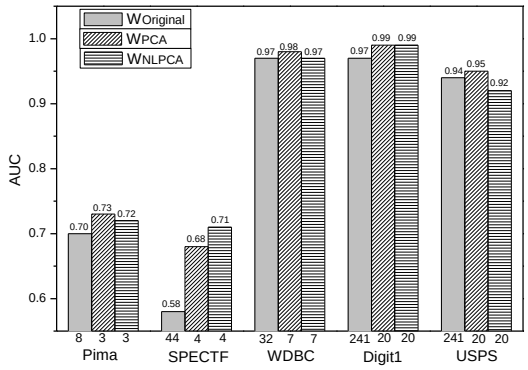

Fig. 6 AUCs for the Five Real-world Data Sets: The number below the individual histogram stands for the number of the features used for making the similarity matrix W.

TABLE I
WILCOXON SIGNED-RANKS TEST

Data set	WOriginal	WPCA		WNLPCA	
	mean ±std	mean ±std	p-value	mean ±std	p-value
Pima	0.70 ±0.037	0.73 ±0.033	0.00	0.72 ±0.035	0.06
SPECTF	0.58 ±0.100	0.69 ±0.129	0.00	0.72 ±0.108	0.00
WDBC	0.97 ±0.010	0.98 ±0.011	0.09	0.97 ±0.014	0.70
Digit1	0.97 ±0.007	0.99 ±0.003	0.00	0.99 ±0.004	0.00
USPS	0.94 ±0.012	0.95 ±0.007	0.00	0.93 ±0.027	0.00

Fig. 7 shows the results. The CF of W_{PCA} or W_{NLPCA} is relatively much higher than that of $W_{Original}$ for each of the five problems. For Pima Indian Diabetes, the CF of W_{PCA} or W_{NLPCA} is higher by three times than that of $W_{Original}$. Similarly, the efficacy of a feature increased for other problems: 12, 5, 12, and 12 times for SPECTF, WDBC, Digit1 and USPS, respectively. This implies that PCA or NLPCA extracts efficacy-enhanced features out of irrelevant (or noise) ones, which consequently enables us to achieve a similar or better performance with a much smaller set of features.

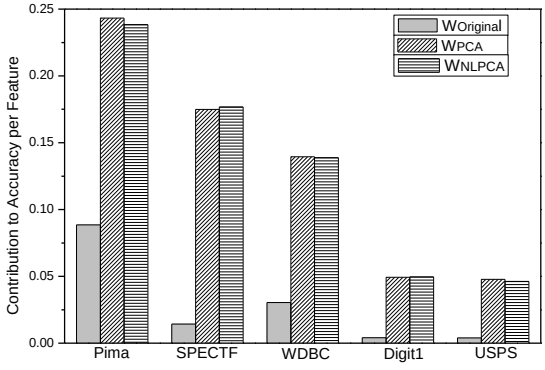

Fig.7 The CF of a Feature

V. CONCLUSION

In this paper, we addressed an issue on how robust the SSL algorithms against the noisy and high dimensional data. Both factors can incur negative influence on the similarity matrix, and consequently, degrade the generalization ability of SSL. As a method of overcoming the difficulty, we proposed to use the extracted features from PCA or NLPCA when calculating the similarities between data points. In the experiment, we showed how much loss of performance can occur because of the noisy and high dimensional features, and presented how to regain or recover the original performance through PCA or NLPCA.

As a preliminary step, the current work takes very simple conventional feature extraction methods as its basis. However, incorporated into more sophisticated state-of-the-art feature extraction algorithms or newer design of a wrapper algorithm for SSL, our approach has the potential to improve considerably on the original performance of SSL and enhance its robustness as well.

ACKNOWLEDGEMENT

This work was supported in part by Post Brain Korea 21 and the research grant form Korean Government (MOEHRD, Basic Research Promotion Fund, KRF-2008-531-D00032).

REFERENCES

[1] Chapelle O., Weston J., and Schoelkopf B.. Cluster kernels for semi-supervised learning. In: *Advances in Neural Information Processing Systems*, edited by S. Becker, S. Thrun, and K. Obermayer, 15, pages 585-592, MIT Press, 2003.

[2] Zhu X.. Semi-supervised learning with graphs. *Ph.D. dissertation*, Carnegie Mellon University, Pennsylvnia, USA, 2005.

[3] Shin H., Hill N.J., and Raetsch G.. Graph-based semi-supervised learning with sharper edges. *Proceeding of the 17th European Conference on Machine Learning*, pages 402-413, 2006.

[4] Shin H., Lisewski A.M. and Lichtarge O.. Graph sharpening plus graph integration: a synergy that improves protein functional classification. *Bioinformatics*, 23(23), pages 3217-3224, Oxford University Press, 2007.

[5] Su W., Carpuat M., and Wu D.. Semi-supervised training of a kernel PCA-based model for word sense disambiguation. *Proceedings of the 20th International Conference on Computational Linguistics*, pages 29-35, Switzerland, 2004.

[6] Shin H., Tsuda K.. Prediction of protein function from networks. In book: *Semi-Supervised Learning*, edited by O. Chapelle, B. Schoelkopf, A. Zien, 20, pages 339-352, MIT press, 2006.

[7] Guyon I., Elisseeff A.. An introduction to variable and feature selection. *Machine Learning Research*, 3, pages 1157--1182, 2003.

[8] Scherf M., Brauer W.. Feature selection by means of a feature weighting approach. *Technical report*, Techniche Universität München, 1997.

[9] Setiono R., Liu H.. Neural-network feature selector. *IEEE Transactions on Neural Networks*, 8, pages 654-662, 1997.

[10] Smith, L.I.: A Tutorial on Principal Components Analysis. 2002

[11] Cao L.J., Chong W.K.. Feature extraction in support vector machine: a comparison of PCA, KPCA and ICA. *Proceedings of the 9th International Conference on Neural Information Processing*, 2, pages 1001-1005, 2002.

[12] Saegusa R.. A study of nonlinear principal component analysis using neural networks. Waseda University, 2005.

[13] Kramer M.A.. Nonlinear principal component analysis using autoassociative neural networks. *AICHE*, 37(2), pages 43-49, 1991.
[14] Ikbal M.S., Misra H., and Yegnanarayana B.. Analysis of autoassociative mapping neural networks. *Neural Networks*, 5, pages 3025-3029, 1999.
[15] Pokajac D., Lazarevic A.. Applications of unsupervised neural networks in data mining. *Neural Network Applications in Electrical Engineering*, pages 17-20, 2004.
[16] Hsieh W.W.. Nonlinear principal component analysis of noisy data. *Neural Networks*, 20(4), pages 434-443, 2007.
[17] Kamimur R.T., Bicciato S., Shimuzu H., and etc. Mining of multivariate temporal biological data: a framework for the rational design of data-driven models. *BioKDD*, San Francisco, CA, 2001.
[18] http://archive.ics.uci.edu/ml/
[19] http://www.kyb.tuebingen.mpg.de/ssl-book/benchmarks.html
[20] Zhu M., Ghodsi A.. Automatic dimensionality selection from the scree plot via the use of profile likelihood. *Computational Statistics& Data Analysis*, 51, pages 918-930, 2006
[21] Demsar, J.. Statistical comparisons of classifiers over multiple data sets. *Machine Learning Research*. 7, pages 1-30, 2006