4-9-2013

Summed Component Analysis for Dimensionality Reduction and Classification

Mopelola Sofolahan
School of Electrical and Computer Engineering, *Purdue University*, msofolah@purdue.edu

Okan Ersoy
School of Electrical and Computer Engineering, *Purdue University*, ersoy@purdue.edu

Summed Component Analysis for Dimensionality Reduction and Classification

Mopelola Sofolahan

Okan Ersoy

TR-ECE-13-05

April 9, 2013

Purdue University

School of Electrical and Computer Engineering

465 Northwestern Avenue

West Lafayette, IN 47907-1285

Summed Component Analysis for Dimensionality Reduction and Classification

Mopelola Sofolahan
School of Electrical and
Computer Engineering
Purdue University
West Lafayette, Indiana 47909–2035
Email: msofolah@purdue.edu

Prof. Okan Ersoy
School of Electrical and
Computer Engineering
Purdue University
West Lafayette, Indiana 47909–2035
Email: ersoy@ecn.purdue.edu

Abstract

In the area of dimensionality reduction, principal component analysis (PCA) has been used with much success. Other dimensionality reduction techniques have been proposed such as principal feature analysis (PFA) which was developed by Ira Cohen, Qi Tian et.al. PFA uses k-means clustering with the principal components to determine principal features. We present a new approach to dimensionality reduction of features called Summed Component Analysis (SCA). SCA uses similar criteria as PFA and PCA to create a lower dimensional feature space. However, it is unique in the way the features in the new space are formed by summing selected features from the original space. The simplicity of the approach lends some advantages to analysis since the new features are simply sums of a selected number of the original features in the lower dimensional space. Furthermore, with SCA we are able to show improved classification performance over PCA, which is known to give impressive lower-dimensional representation of a dataset, but which doesn't always translate to improvement in classification. SCA can prove useful when applied to high dimensional data sets to be classified, such as physical measurements that describe different scenarios, or in the area of financial data analysis in which different stocks are to be combined in a way that provide optimal information needed to classify stock market trends.

I. Introduction

In the analysis of high dimensional data, it is necessary to project data from a higher dimensional space to a lower dimensional space while retaining as much of the relevant information in the data as possible. Many approaches have been taken to dimensionality reduction, such as principal component analysis (PCA), independent component analysis (ICA), and principal feature analysis (PFA). PCA aims to represent data in a lower dimensional space with new orthogonal features, the principal components, formed as weighted combinations of the original features. With ICA, the the goal is to determine a linear representation of nongaussian data so that the components are statistically independent, or as independent as possible [1]. Finally, with PFA, the goal is to select a subset of the original features rather than finding a mapping that uses all the original features [2], [3]. These approaches to dimensionality reduction suffer from some drawbacks. With PCA and ICA, it is difficult to assign any physical interpretation to the features in the new space. PFA does not suffer from this drawback, since only a subset of the features which retain their physical meanings are selected. However, it results in a loss of information as majority of the features are discarded. Furthermore, with PCA, while the transformation gives a new feature space that is selected to make the greatest contribution to representation of the original data, this does not always lead to improvement in classification [4].

In this work, we propose an efficient method that exploits the structure of PCA and PFA to find a lower k-dimensional space which makes use of all the original features by first dividing the features into k non-overlapping groups which are each summed to create k new features.

The paper is organized as follows. We begin with a brief review of PCA and PFA and describe how SCA deviates from these two approaches in Section II. Next we apply comparatively the method of SCA and PCA to Wisconsin Diagnostic Breast Cancer data [5], and to synthetically generated Gaussian mixture data and describe the performance metrics used in Section III. We then compare the classification accuracies using these two different methods and compare the class scatter performance in the lower dimensional SCA and PCA space in Section IV. Finally, we conclude with a discussion of our results and future work to be done.

II. Theory

To transform a set of random observations in a matrix $\mathbf{X} \in \mathbf{R}^{n\times m}$ to a lower dimensional space $\mathbf{Y} \in \mathbf{R}^{k\times m}$ using SCA, we begin with the steps used in PCA. Given that $\mathbf{X}$ has zero mean, with rows corresponding to features and columns corresponding to observations, the transformation matrix $\mathbf{T}$ that takes $\mathbf{X}_{n\times m} \rightarrow \mathbf{Y}_{\mathbf{PCA}n\times m}$ is obtained by finding the transpose of the orthogonal matrix $\mathbf{A}$ formed as a result of arranging the eigenvectors of $\sum_x$, the covariance matrix of $\mathbf{X}$, in decreasing order of their associated eigenvalues. Given the eigen-decomposition of $\sum_x$,

$$\sum\nolimits_x = \mathbf{A}\Lambda\mathbf{A}' \tag{1}$$

where Λ is a diagonal matrix with diagonal elements being the eigenvalues of $\sum_x$, $\lambda_1 \geq \lambda_2 \geq ...\lambda_n$. The columns of $\mathbf{A}$ are the eigenvectors of $\sum_x$. The PCA transformation is given by

$$\mathbf{Y}_{PCA} = \mathbf{A}'\mathbf{X} \tag{2}$$

where the transormation matrix $\mathbf{T} = \mathbf{A}'$. Typically, a lower-dimensional subspace, k is desired. To obtain this, only the first k eigenvectors in $\mathbf{A}$ are used to form the transormation matrix.

Let us designate the rows of the matrix $\mathbf{A}$ as $v_1, v_2, \ldots, v_n \in \mathbf{R}^n$. The elements of each vector v_i correspond to the weights used with the ith feature of $\mathbf{X}$ to generate the feature in the PCA space [2]. PFA makes use of the property that features which are highly correlated will have similar weight vectors v_i while features that are uncorrelated will have weight vectors that are quite different [2]. With PFA, the subset of features is chosen by clustering the v_i vectors into k clusters to determine those features which are highly correlated, and then choosing one feature from each set - the feature used with the vector which is closest to the mean of the cluster. The idea behind this is that the vector v_i closest to the mean vector of the cluster represents the cluster best and points to a corresponding input feature. This is believed to be a good representation of the original data [2], [3].

In SCA, we perform the same steps as PCA and PFA to obtain the orthogonal matrix $\mathbf{A}$. The rows of this matrix are similarly labeled as $v_1, v_2, \ldots, v_n \in \mathbf{R}^n$, and we perfrom k-means clustering algorithm to cluster the vectors. The key difference between SCA and PFA is that we next retain all the features in SCA, as described in the steps below.

1) Compute the matrix $\mathbf{A}$ as defined in equation (1).
2) Keeping the entire n dimensional space for $\mathbf{A}$, i.e., retaining all the eigenvectors of $\sum_x$, we find k clusters of the vectors $v_1, v_2, \ldots, v_n$ using the k-means clustering algorithm with Euclidean distance as the distance metric.
3) For the first cluster, replace the vectors in the cluster with the vector w_1 which is the vector that is closest to the center of the first cluster. Repeat this step for the remaining $k-1$ clusters. This results in an approximation to the matrix $\mathbf{A}$, which we denote as $\tilde{\mathbf{A}}$.
   For instance, if $k = 2$ clusters, let us assume that v_1, v_2, v_3 are assigned to cluster 1, and replaced with center vector w_1; $v_4, v_5, \ldots, v_n$ are assigned to cluster 2 and replaced with center vector w_2. $\tilde{\mathbf{A}}$ can be expressed as

$$\tilde{\mathbf{A}} = \begin{bmatrix} \cdots\cdots w_1 \cdots\cdots \\ \cdots\cdots w_1 \cdots\cdots \\ \cdots\cdots w_1 \cdots\cdots \\ \cdots\cdots w_2 \cdots\cdots \\ \cdots\cdots w_2 \cdots\cdots \\ \vdots \\ \cdots\cdots w_2 \cdots\cdots \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ \vdots & \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cdots\cdots w_1 \cdots\cdots \\ \cdots\cdots w_2 \cdots\cdots \end{bmatrix} \tag{3}$$

   Let us define the matrix consisting of the vectors closest to each cluster center, $\mathbf{W}$ as

$$\mathbf{W} = \begin{bmatrix} \cdots\cdots w_1 \cdots\cdots \\ \cdots\cdots w_2 \cdots\cdots \end{bmatrix} \tag{4}$$

   Therefore, $\mathbf{Y}_{SCA}$ which is an approximation to $\mathbf{Y}_{PCA}$, can be expressed as

$$\mathbf{Y}_{SCA} = \tilde{\mathbf{A}}'\mathbf{X} \tag{5}$$

$$= \begin{bmatrix} \vdots & \vdots \\ w_1{}' & w_2{}' \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \cdots\cdots X_1 \cdots\cdots \\ \cdots\cdots X_2 \cdots\cdots \\ \cdots\cdots X_3 \cdots\cdots \\ \cdots\cdots X_4 \cdots\cdots \\ \cdots\cdots X_5 \cdots\cdots \\ \vdots \\ \cdots\cdots X_n \cdots\cdots \end{bmatrix} \tag{6}$$

$$= \begin{bmatrix} \vdots & \vdots \\ w_1{}' & w_2{}' \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} \cdots\cdots X_{SCA_1} \cdots\cdots \\ \cdots\cdots X_{SCA_2} \cdots\cdots \end{bmatrix} \tag{7}$$

   Thus, $\mathbf{X}_{SCA}$, is obtained by summing the features in $\mathbf{X}$, which correspond to the vectors in each cluster. In general, for k clusters, $\mathbf{X}_{SCA}$ would be of the form

$$\mathbf{X}_{SCA} = \begin{bmatrix} \cdots\cdots X_{SCA_1} \cdots\cdots \\ \cdots\cdots X_{SCA_2} \cdots\cdots \\ \cdots\cdots X_{SCA_3} \cdots\cdots \\ \vdots \\ \cdots\cdots X_{SCA_k} \cdots\cdots \end{bmatrix} \tag{8}$$

Note that $\mathbf{X}_{SCA}$ consists of k derived features, which are obtained as the sums of features that are found to be similar to each other based on the clustering algorithm. Thus, unlike PFA, we make use of all the original features of $\mathbf{X}$, and we do so by summing the features in each cluster.

III. Experiments

We experimented with the SCA algorithm using two different datasets. The first is The Wisconsin Diagnostic Breast Cancer (WDBC) [5] data set which consists of 30 measurements (features) and corresponding diagnosis for 569 patients. With this dataset, we want to predict the correct class (one of two options) for each patient using the given measurements.

The second dataset is a synthetically generated mixture of Gaussian samples with different probability density functions. We use the `gmm` function [6] to generate the synthetic data having 3 classes, and 2100 observations of 20 features each. The prior class probabilities are set to be equal, all classes have different means, and the covariance values of class 1 and class 3 are set equal just for increased difficulty.

Fig. 1 and Fig. 2 below show 3D scatterplots of the datasets. The three axes are randomly selected features for illustration purposes.

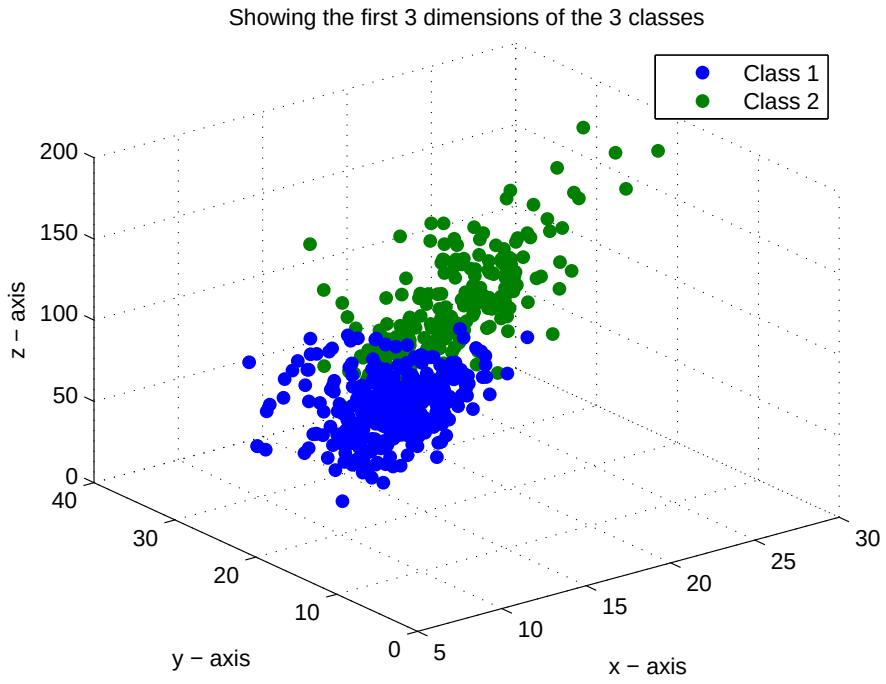


Fig. 1. Scatterplot of WDBC Data with the first 3 dimensions.

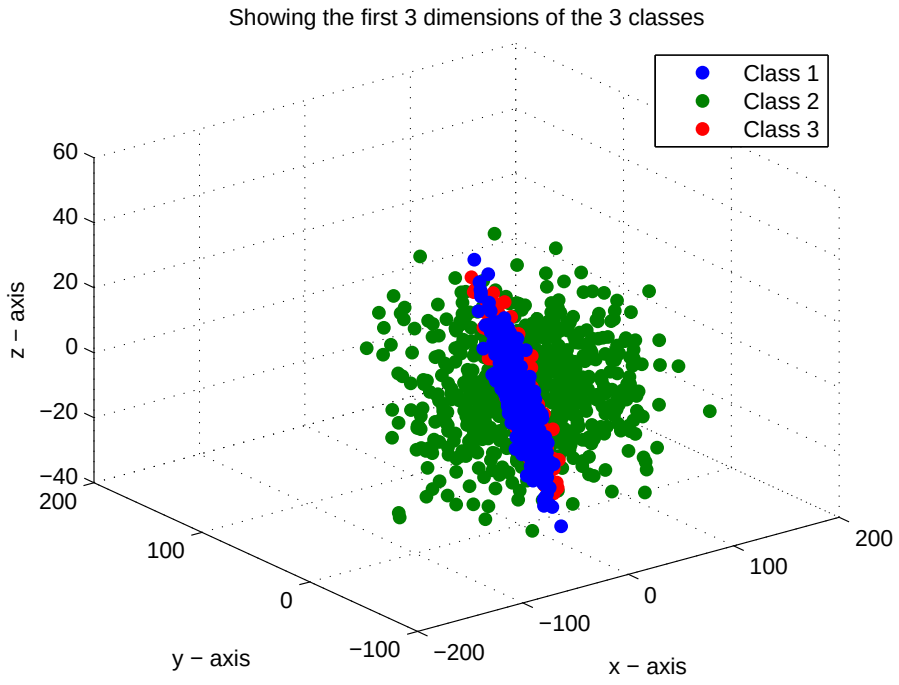


Fig. 2. Scatterplot of Gaussian Data with the first 3 dimensions.

To implement SCA, we begin by dividing each dataset into three groups to be used for training, validating, and testing. SCA is then computed with the training set for k, the number of clusters, ranging from 1 to the maximum number of features. The validating and testing sets are transformed to the new SCA space by summing the features in the same manner as with the training set. A quadratic maximum likelihood classifier is used to determine the separating hyperplanes for classifying the data points in the training set, and the same classifier is used with the validating and testing sets in the SCA domain.

To evaluate the performance of SCA, we determine the similarity between classes that are represented in the SCA space by using scatter matrices. We evaluate the trace of the between-cluster ($\mathbf{S}_B$) to within-cluster ($\mathbf{S}_W$) scatter ratio $\mathtt{tr}[\mathbf{S_W}^{-1}\mathbf{S_B}]$, with large values indicating good partition of the data classes [7].
The between-class scatter matrix is computed as:

$$\mathbf{S}_B = \sum_{i=1}^{c} n_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^t \tag{9}$$

The within-class matrix is computed as:

$$\mathbf{S}_W = \sum_{i=1}^{c} \sum_{\mathbf{x}\in D} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t \tag{10}$$

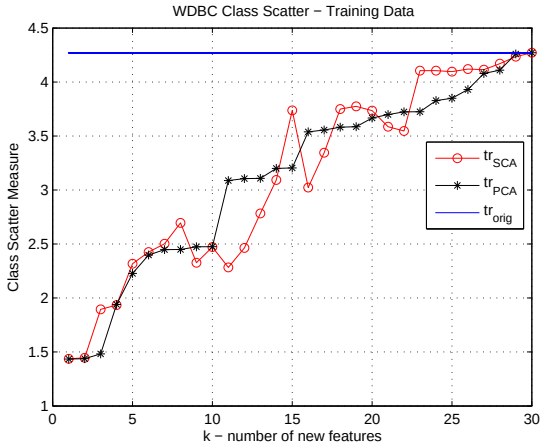
where
n_i: number of samples in class i
$\mathbf{m}_i$: mean of class i
$\mathbf{m}$: total mean vector
c: number of classes

The second metric used for evaluating the performance of SCA is the classification accuracy.

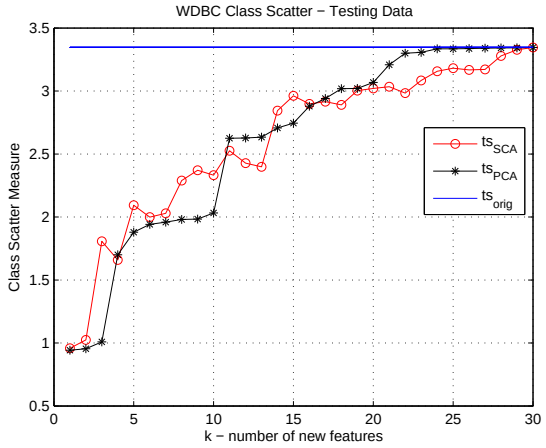
IV. Results

We compare the results of classification accuracy with the original datasets and the datasets after they have been pre-processed with SCA and PCA.

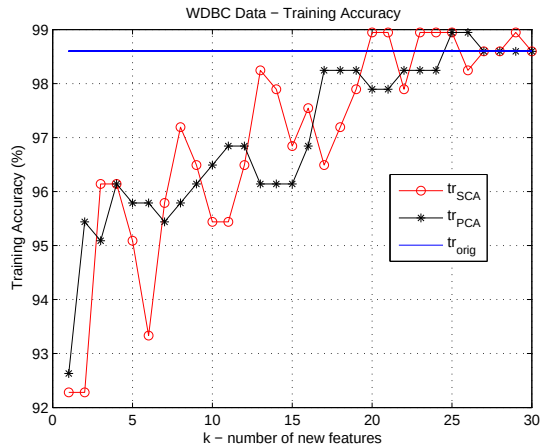
Figures below show plots of the class scatter and testing accuracies using the WDBC dataset and synthetic Gaussian data.



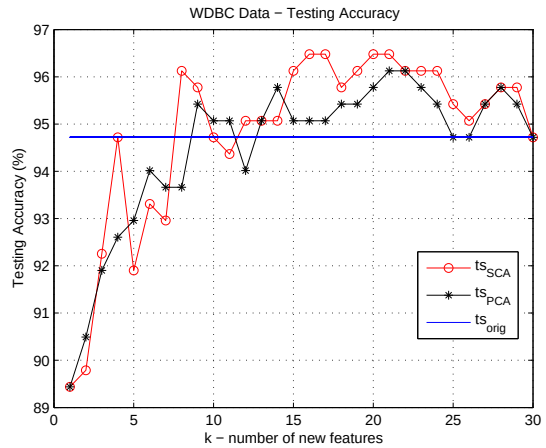
(a) Training Class Scatter.



(b) Testing Class Scatter.

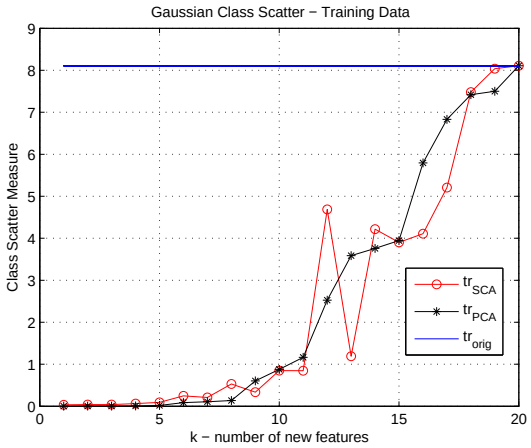


(c) Training Classification Accuracy.

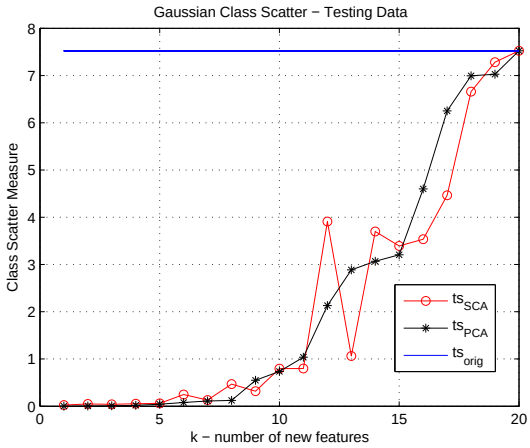


(d) Testing Classification Accuracy.

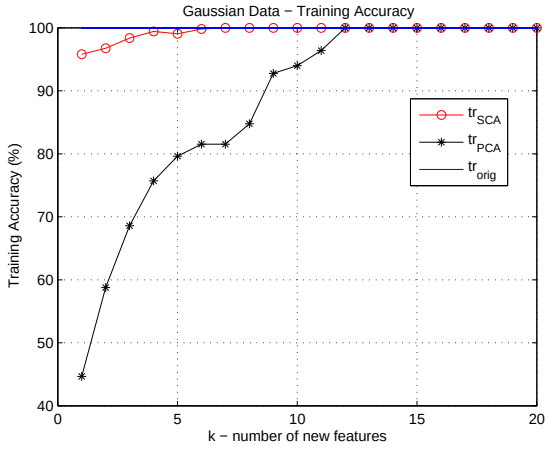
Fig. 3. WDBC Data: Class Scatter and Classification Accuracy for Training and Testing Sets.



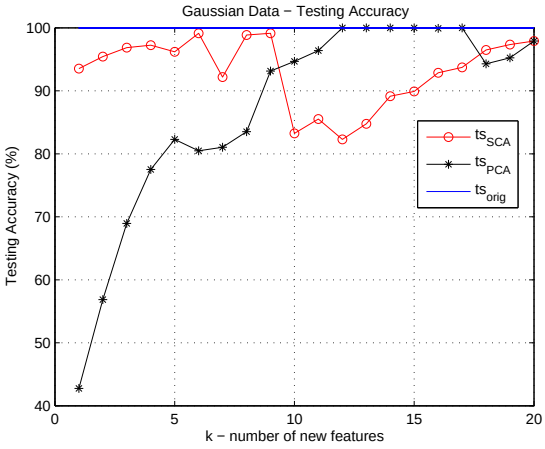
(a) Training Class Scatter.



(b) Testing Class Scatter.



(c) Training Classification Accuracy.



(d) Testing Classification Accuracy.

Fig. 4. Gaussian Data: Class Scatter and Classification Accuracy for Training and Testing Sets.

From FIG 3(a) and FIG 3(b), we see that the class separation in the SCA space and the PCA space are comparable over the range of possible k values. In particular, with SCA, the separation between classes appears larger most of the time when $k \leq$10. Similar observations can be made about the class separation in the SCA space for the synthetic Gaussian data as shown in FIG 4(a) and FIG 4(b).

In addition, for the synthetic Gaussian data, classification is far better in the SCA space than in the PCA space. During testing, the classification accuracy using SCA is much higher for $1 \leq k \leq 9$. Thus, for classification in a lower dimensional space, SCA would be the better choice with this dataset. On the other hand, for the WDBC data, the classification results are not as consistent. Although the accuracies are still shown to be comparable with those obtained in the PCA space, they are not consistently higher with SCA. With this dataset however, we are able to obtain testing accuracies that are higher using $12 \leq k <$30. Thus, it is still useful to perform dimensionality reduction on this dataset prior to classification, even though we cannot categorically state that either SCA or PCA gives better classification performance here.

V. Conclusions

We have shown a new approach to dimensionality reduction of features using summed component analysis. SCA has the advantages that features in the SCA space have a simpler conceptual meaning to the user, and also no information is lost during the transformation. Using the example datasets, it is clear that SCA is a viable competitor with PCA in representing data at lower dimensions. In the area of classification, SCA has been shown to outperform PCA with certain datasets, and to be at least comparable in performance to PCA with other datasets.

In future work, we will investigate the effects of reducing the dimensionality of the rows of $\mathbf{A}$ prior to clustering of features. We will also consider using other similarity measures to group features before they are summed and observe how these affect performance. Finally, we will broaden our pool of datasets to include others which are more difficult to classify and work with more complex classifiers.

REFERENCES

[1] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[2] I. Cohen, Q. T. Xiang, S. Zhou, X. Sean, Z. Thomas, and T. S. Huang, "Feature selection using principal feature analysis," 2002.

[3] T. H. Ling, N. Chaudhari, and Z. Junhong, "Time series prediction using principal feature analysis," in *Industrial Electronics and Applications, 2008. ICIEA 2008. 3rd IEEE Conference on*, June, pp. 292–297.

[4] D. A. Landgrebe, *Signal Theory Methods in Multispectral Remote Sensing*. Newark, NJ: Wiley, 2003.

[5] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[6] I. T. Nabney, *NETLAB: algorithms for pattern recognition*. New York, NY, USA: Springer, 2002.

[7] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York, NY, USA: Wiley, 2001.