A PCA/ICA BASED FEATURE SELECTION METHOD AND ITS APPLICATION FOR CORN FUNGI DETECTION

Zehra Cataltepe, Hakki Murat Genc**, Thomas Pearson****

* Istanbul Technical University, Computer Engineering Department, Istanbul, Turkey
cataltepe@itu.edu.tr

** Information Technologies Institute, Marmara Research Center,
The Scientific and Technological Research Council of Turkey, Kocaeli, Turkey
murat.genc@bte.mam.gov.tr

*** Engineering Research Unit, USDA-ARS, Manhattan, KS, USA
thomas.pearson@gmprc.ksu.edu

ABSTRACT

Dimensionality reduction algorithms help reduce the classification time and sometimes the classification error. For time critical applications, in order to have reduction in the feature acquisition phase, feature selection is preferable to dimensionality reduction, which requires measurement of all inputs. Traditional feature selection methods, such as forward or backward selection, are costly to implement. We introduce a new feature selection method that decides on features to retain, based on how PCA (Principal Component Analysis) or ICA (Independent Component Analysis) values them. We compare the accuracy of our method to PCA and ICA using the same number of principal/independent components. We also do comparison to backward and forward selection with the same number of features. For our experiments, we use spectral measurement data taken from corn kernels infested and undamaged by fungi. Our algorithm selects features with almost as good classification accuracy as forward/backward selection and is a lot faster than those algorithms. It also results in better classification accuracy then using the same number of principal/independent components.

1. INTRODUCTION

Feature selection methods (for example, [10, 2, 4, 11, 5]) are often used in pattern recognition applications. These methods choose a number of features among the original features. An obvious advantage of using feature selection is reduction in the time and cost of feature acquisition, as well as reduction in classifier training and testing time. Feature selection is also helpful in improving classifier accuracy, provided that noisy, irrelevant or redundant features are eliminated.

Forward and backward feature selection algorithms are useful in identifying a good set of features. However, they require training and testing a classifier at each feature addition/elimination step, hence are very costly in terms of time. PCA (Principal Component Analysis) and ICA (Independent Component Analysis) [6] have widely been used for dimensionality reduction. While PCA works best when the input distribution is gaussian, ICA works best for nongaussian data.

In this study, we introduce a feature selection method that relies on how PCA and ICA (FastICA implementation [6]) values a feature to eliminate a feature. The method starts with all features and reduces them one by one, and hence is similar to backward selection. It is much faster than backward selection since the feature evaluation is made based on PCA, which is usually much faster than training a classifier. Previously [11] has used entropy based symmetric uncertainty to measure the relevance and redundancy of each feature. Although [11] introduces a quite fast algorithm, it eliminates each feature it deems irrelevant and may miss features that could have been useful when used together. The symmetric uncertainty is also used in [5], to measure the degree of association between features.

The rest of the paper is organized as follows: In Section 2 we introduce the corn spectra data that we use in our experiments. In Section 3 we summarize the feature selection and dimensionality reduction algorithms we considered. In Section 4 we introduce our feature selection algorithm based on PCA/ICA. Section 5 includes information on the accuracy and timing results we obtained for each method. Section 6 concludes the paper.

2. CORN DATA

We used data from the agricultural domain, corn kernels infected and un-infected with certain fungi [7], to perform experiments on our algorithm. In [7], in addition to single-kernel reflectance spectra (550 to 1700 nm), visible color reflectance images, x-ray images, multi-spectral transmittance images (visible and NIR), and physical properties (mass, length, width, thickness, and cross-sectional area) were also used to determine if they could be used to detect fungal-infected corn kernels. Kernels were collected from corn ears inoculated with one of several different common fungi (*Aspergillus flavus, Aspergillus niger, Diplodia maydis, Fusarium graminearum, Fusarium verticillioides,* or *Trichoderma viride*) several weeks before harvest, and then collected at harvest time. Authors found that using a neural network and two NIR reflectance spectral bands centered at 715 nm and 965 nm, they could correctly identify more than 95% of both asymptomatic kernels and kernels showing extensive discoloration. They also note that these two spectral bands can easily be implemented on high-speed sorting machines for removal of fungal-damaged grain.

In this study, we only used the single-kernel reflectance spectra (550 to 1700 nm) data. There were a total of $N = 1648$ data points consisting of $d = 241$ dimensional feature vectors. We concentrated on the problem of differentiating

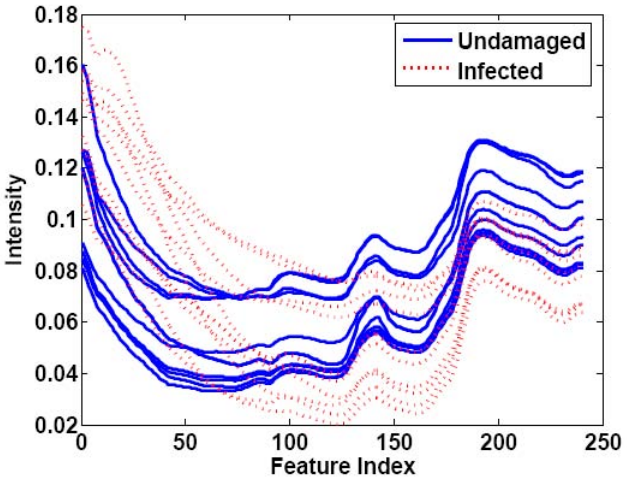

Figure 1: Sample measurements from infected and undamaged corn kernels.

infected and undamaged kernels from each other, hence we had a binary classification problem.

In the corn kernel dataset, the spectrometer measures the absorbance of the plant. The reflectance spectra of the corn kernels between 550 and 1700nm are the 241 features. A set of measurements for infected and undamaged kernels are shown in Figure 1. At the first glance to Figure 1 the 'red's and 'blue's are most separable at the beginning and at the end of the reflectance spectra. That is to say the features [1st - 15th] (corresponding to 550nm to 625nm) and [200th - 241th] (corresponding to 1500nm to 1700nm) are the best choices.

The covariance matrix (not shown due to insufficient space) also has its biggest diagonal values at regions [200th feature - 241th feature] and [1st feature - 15th feature] and [175th feature - 185th feature] respectively. The performance results for selecting 5 and 20 features from the most distinctive region ([200th feature - 241th feature]) are compared with the same size PCA using Fisher's Linear Discriminant in Table 1. 5 (or 20) features are randomly selected from the given feature interval and mean value of the 10 fold cross validation error is reported.

In many applications, the data is preprocessed before feature selection/dimensionality reduction. While this spectral data do not need normalization, smoothing operation improves the separability. We think this is because the neighbors carry information in spectral datasets. Another advantage of the smoothing process is the elimination of the outliers, which was of secondary importance for us since the dataset does not have much off-mean elements. The improvement with smoothing is obvious from error reports. (Table 1.) These results are obtained by using a gaussian smoothing filter (FWHM filter) which values the closer neighbors more.

3. FEATURE SELECTION ALGORITHMS

Forward and backward feature selection and PCA (Principal Component Analysis) and ICA (Independent Component Analysis) [6] are the most commonly used feature selection and dimensionality reduction algorithms [3].

Backward and feature selection are wrapper methods, where each feature is checked to be included or excluded after a classifier is trained and then tested or validated. For large dimensional spaces these algorithms become impossible to run. Forward feature selection also suffers from the fact that features that could be useful when together but useless when alone are missed.

Both PCA and ICA project the d dimensional input space to a lower dimensional space d'.

The goal of ICA is to find a linear representation of non-gaussian data so that the components are statistically independent, or as independent as possible [6]. While the other well known linear transformation methods (e.g. PCA) benefit from the gaussianity of the data, ICA improves the classifier performance in the opposite case.

In order to simply formulate what ICA does, let us assume n dimensional observation vectors $x = (x_1, x_2, \ldots, x_m)^t$ which are zero mean random variables. Let $s = (s_1, s_2, \ldots, s_{d'})^t$ be the d'-dimensional transform of x. Then the problem is to determine a constant weight matrix W so that the linear transformation of the observed variables

$$s = Wx \tag{1}$$

has certain properties. In other words, the observed signal x can be written in terms of the independent components:

$$x = A^{-1}s \tag{2}$$

where A is the inverse (or the pseudo-inverse) of the W transform matrix.

PCA (Principal Component Analysis) also uses a transform matrix W. However the entries of W are the eigenvectors of the covariance matrix of the inputs. Only the eigenvectors corresponding to the largest d' eigenvalues are selected.

4. PCA/ICA BASED FEATURE SELECTION

Both the PCA Based Dimensionality reduction and the ICA Based Dimensionality reduction algorithms are based on the idea that the features that are least important are the ones whose contribution to the principal/independent components are the least. The least important feature(s) are eliminated and the principal/independent components are recalculated based on the remaining features. The degree of contribution of a feature is approximated as the sum of the absolute values of the transform matrix W entries associated with the feature.

Let d be the input dimensionality, d' be the PCA/ICA reduced space's dimensionality and d^* be the number of features to be selected. δd shows the number of features to be eliminated at each step. By default and in the experiments reported below, $\delta d = 1$, however this value could be increased to make our feature selection faster. Initially d is assigned the original input dimension. We give the steps of our algorithm below:

- Calculate the PCs/ICs. Find the transform matrix W such that $Z = A * W$ and where $Z_{Nxd'}$ is the matrix of PCs/ICs, A_{Nxd} are the original inputs and $W_{dxd'}$ is the transform matrix.
- Sum the absolute values of entries for each row of W. Each row of W matrix multiplies with the feature vector. If the entries are 'small' for any row of the W matrix, the contribution of the corresponding feature to the principal/independent components is 'small'.

- Find the minimum δd sums and eliminate the corresponding features. $d = d - \delta d$.
- If $d > d^*$ (there are still more features than needed) go back to the first step, using the features remained after the elimination.

While the ICA algorithm optimizes the number of independent components at each step of feature elimination, the number of principal components can be chosen at the very first step. The only limitation is that the number of principal components must be less then or equal to the number of features to be retained.

5. RESULTS

5.1 Classifiers Used

Two different classifiers, Fisher's linear classifier and logistic linear classifier, are used to test the proposed algorithm [1] [3, 8].

Fisher's linear classifier finds the linear discriminant function between the classes in the dataset by minimizing the errors in the least square sense. Logistic Linear classifier performs the computation of the linear classifier for the dataset by maximizing the likelihood criterion using the logistic (sigmoid) function. In logistic discrimination, rather than modeling the class-conditional densities, we model their ratio. [1].

5.2 Experimental Results

We report the experimental results for the original and preprocessed spectra data. Tables 2 and 3 show the mean 10-fold cross validation accuracies for the original input features. Tables 4 and 5 show the results when the data is preprocessed. Preprocessed data is obtained by smoothing with FWHM (full width at half maximum) pass-band interference filters. In both cases, the feature vectors are not normalized to zero mean and unit variance, because there is a lot of correlation between neighboring feature components and normalization could loose this information.

We compare our PCA and ICA based algorithms' accuracy to that of backward and forward feature selections and to PCA/ICA. Our methods sometimes outperform the backward/forward selection. Note that for very large datasets, forward/backward feature selection may not be feasible. So, as the input space dimensionality grows, PCA/ICA based dimensionality reduction algorithm becomes a stronger alternative to forward/backward selection.

Our method performs better than PCA or ICA when the same number of principal/independent components are used. Note that for the real time corn detection problem, instead of dimensionality reduction provided by PCA/ICA, feature selection provided by our method would be preferable.

We also compare the speed of our algorithm to that of backward feature selection in Table 6 when Fisher's linear discriminant classifier is used. The results are reported in seconds and are measured on a 3.2 P4 dual core processor with 2Gb Ram where one processor is dedicated for each time consumption measurement case. Our algorithm is a lot faster than backward feature selection.

[1] Although we did experimented with the support vector classifier, we were not able to get results in a reasonable time for our data set, hence we could not include it here.

6. CONCLUSIONS AND FUTURE RESEARCH

We presented a fast and accurate feature selection algorithm that is based on PCA/ICA. Our algorithm selects features that result in as good classifier accuracy as forward/backward selection and is a lot faster than those algorithms. Although, for 271 dimensional corn data backward/forward selection algorithms could be implemented offline, these algorithms become infeasible for very large (thousands of) dimensional datasets. The features selected also result in better classification accuracy then using the same number of principal/independent components.

In the near future, we are planning to experiment on improving the speed of our algorithm by eliminating a number of least important features as opposed to a single feature at a time (see δd in Section 4). Time complexity analysis of the algorithm is also among our feature work. The current version of the algorithm requires the inputs to be positive. An extension to the more general inputs case is necessary so that we can test the algorithm on other data sets, such as, for example, UCI Machine Learning Repository data.

REFERENCES

[1] E. Alpaydin, *Introduction to machine learning*. Cambridge, Massachusetts: The MIT Press, 2004.

[2] J. Bins and B. A. Draper, "Feature selection from huge feature sets," in *Eighth International Conference on Computer Vision (ICCV 01),* 159-165, 2001.

[3] R. Duda, P. Hart and D. Stork, *Pattern classification, 2nd ed.* New York: John Wiley and Sons, 2001.

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

[5] M. A. Hall, "Correlation-based feature selection for discrete and numeric class machine learning," in *the Seventeenth International Conference on Machine Learning ,* 159–165, 2000.

[6] A. Hyvarinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Networks*, 13, 411-430, 1999.

[7] T. C. Pearson and D. T. Wicklow, "Properties of corn kernels infected by fungi," *Transactions of the ASAE*, 49, 1235-1245, 2006.

[8] S. Raudys and R. Duin, "On expected classification error of the fisher linear classifier with pseudoinverse covariance matrix," *Pattern Recognition Letters*, 19, 385-392, 1998.

[9] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," *Journal of Machine Learning Research*, 5, 1205-1224, 2004.

[10] Y. Yang and J. Pedersen, "A comparative study on feature selection in text categorization," in *International Conference on Machine Learning,* 412-420, 1997.

[11] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," *Journal of Machine Learning Research*, 5, 1205-1224, 2004.

Table 1: Mean validation error for Fisher's Linear Discriminant (No preprocessing).

NUM FEATURES	PCA	RANDOM CHOSEN FROM 200TH-241TH	RANDOM CHOSEN FROM 200TH-241TH (SMOOTH)
5	0.130	0.070	0.064
20	0.076	0.046	0.035

Table 2: Mean validation error for Fisher's Linear Discriminant (No preprocessing).

NO.OF FEATURES	BACKWARD FEATURE SELECT.	FORWARD FEATURE SELECT.	PCA BASED FEATURE SELECT.	ICA BASED FEATURE SELECT.	PCA	ICA
1	0.205	0.205	0.205	0.205	0.266	0.266
3	0.103	0.094	0.134	0.137	0.120	0.120
5	0.067	0.063	0.118	0.129	0.113	0.113
10	0.029	0.038	0.098	0.045	0.099	0.097
20	0.001	0.006	0.035	0.024	0.079	0.079
30	0.001	0.007	0.023	0.017	0.018	NA
200	0.000	0.001	0.001	0.002	0.000	NA

Table 3: Mean validation error for Logistic Linear Classifier (No preprocessing).

NO.OF FEATURES	BACKWARD FEATURE SELECT.	FORWARD FEATURE SELECT.	PCA BASED FEATURE SELECT.	ICA BASED FEATURE SELECT.	PCA	ICA
1	0.205	0.204	0.206	0.205	0.260	0.260
3	0.097	0.087	0.149	0.132	0.133	0.133
5	0.043	0.055	0.121	0.123	0.126	0.126
10	0.018	0.037	0.076	0.038	0.094	0.093
20	0.009	0.018	0.023	0.027	0.057	0.057
30	0.008	0.004	0.024	0.020	0.013	NA
200	0.001	0.001	0.001	0.003	0.000	NA

Table 4: Mean validation error for Fisher's Linear Discriminant (Smooth Data).

NO.OF FEATURES	BACKWARD FEATURE SELECT.	FORWARD FEATURE SELECT.	PCA BASED FEATURE SELECT.	ICA BASED FEATURE SELECT.	PCA	ICA
1	0.277	0.205	0.206	0.205	0.266	0.266
3	0.135	0.093	0.134	0.089	0.119	0.119
5	0.084	0.065	0.112	0.085	0.114	0.114
10	0.026	0.032	0.099	0.066	0.102	0.095
20	0.012	0.021	0.043	0.032	0.082	0.082
30	0.005	0.011	0.018	0.018	0.018	NA
200	0.000	0.000	0.001	0.001	0.000	NA

Table 5: Mean validation error for Logistic Linear Classifier (Smooth data).

NO.OF FEATURES	BACKWARD FEATURE SELECT.	FORWARD FEATURE SELECT.	PCA BASED FEATURE SELECT.	ICA BASED FEATURE SELECT.	PCA	ICA
1	0.205	0.205	0205	0.205	0.260	0.260
3	0.122	0.087	0.140	0.088	0.132	0.132
5	0.046	0.054	0.104	0.083	0.124	0.124
10	0.024	0.030	0.085	0.056	0.092	0.088
20	0.012	0.022	0.030	0.037	0.059	0.059
30	0.010	0.014	0.013	0.019	0.015	NA
200	0.001	0.000	0.000	0.001	0.000	NA

Table 6: Time required for feature selection and training using Fisher's Linear Discriminant.

NO.OF FEATURES	BACKWARD FEATURE SELECT.	PCA BASED FEATURE SELECT.	ICA BASED FEATURE SELECT.
3	7886	182	2834
5	7885	182	2834
10	7883	182	2833
20	7876	182	2828
30	7864	181	2820