

High Dimensional Image Data Classification Using Parametric and Nonparametric Feature Extractions and Classifiers

Bor-Chen Kuo*, Ginn-Min Yang* and David. A Landgrebe**

*Department of Mathematics Education
National Taichung Teachers College, Taichung 403, Taiwan, R.O.C.
E-mail: kbc@mail.ntctc.edu.tw and ygm@ms3.ntctc.edu.tw
Phone: +886-4-22263181 EXT 223

**School of Electrical and Computer Engineering
Purdue University, West Lafayette, Indiana 47907-1285, U.S.A.
E-mail: landgreb@ecn.purdue.edu

ABSTRACT

In this paper, the performances of different combinations of feature extraction and classifiers are compared based on high dimensional image data. The results show that for high dimensional data classification with small training sample size, the new nonparametric weighted feature extraction with Gaussian classifier has the best performance.

1. INTRODUCTION

For classifying high dimensional image data, a useful processing model that has evolved in the last several years [1] is shown schematically in Figure 1. Research has shown that achieving high precision in modeling the desired classes quantitatively is the critical element to the most effective analysis. Given the availability of data (box 1), the process begins by the analyst specifying what classes are desired, usually by labeling training samples for each class (box 2). New elements to class modeling that have proven important in the case of high dimensional data are those indicated by boxes in the diagram marked 3 and 4.

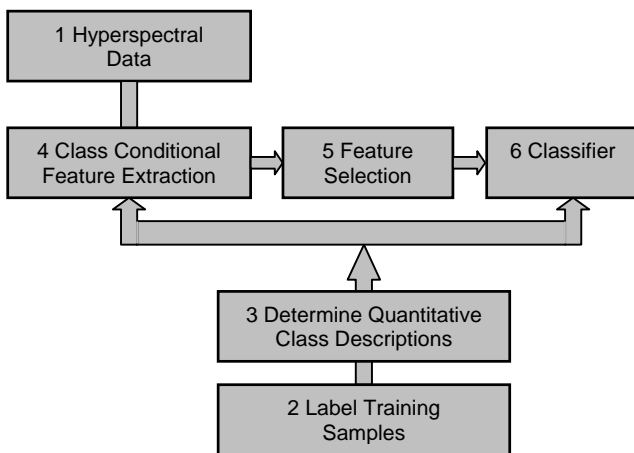


Figure 1. A schematic diagram for classifying high dimensional data

These are the focus of this work and will be discussed in more detail shortly, however the reason for their importance in this context is as follows. Classification techniques

in pattern recognition typically assume that there are enough training samples available to obtain reasonably accurate class descriptions in quantitative form. Unfortunately, the number of training samples required to train a classifier for high dimensional data is much greater than that required for conventional data, and gathering these training samples can be difficult and expensive. Therefore, the assumption that enough training samples are available to accurately estimate the class quantitative description is frequently not satisfied for high dimensional data. Small training sets usually cause Hughes phenomenon [7] and singularity problems. Feature extraction is a usual way to overcome these problems.

The purpose of this paper is trying to find a good combination of feature extraction method and classifier for classifying high dimensional image data. For approaching this purpose, the classification performances using parametric and nonparametric classifiers based on some typical and new feature extraction methods are compared.

2. FEATURE EXTRACTIONS

There are many feature extraction methods, and they can be categorized into two groups, parametric and nonparametric feature extractions [Fuku]. In this section, only those feature extractions, which are used in this study, will be introduced in the followings.

2.1 Parametric Feature Extractions

Linear Discriminant Analysis (LDA) or Discriminant Analysis Feature Extraction (DAFE) is often used for dimension reduction in classification problems. It is also called the parametric feature extraction method in [2], since DAFE uses the mean vector and covariance matrix of each class. The purpose of DAFE is to find a transformation matrix A such that the class separability of transformed data (Y) is maximized. Usually within-class, between-class, and mixture scatter matrices are used to formulate the criteria of class separability. A within-class scatter matrix for L classes is expressed by [2] :

$$S_w = \sum_{i=1}^L P_i E\{(X - m_i)(X - m_i)^T | \omega_i\} = \sum_{i=1}^L P_i \Sigma_i = \sum_{i=1}^L P_i S_{wi}$$

where P_i means the prior probability of class i , m_i is the class mean and Σ_i is the class covariance matrix. A be-

tween-class scatter matrix is expressed as

$$S_b = \sum_{i=1}^L P_i (m_i - m_0)(m_i - m_0)^T = \sum_{i=1}^{L-1} \sum_{j=i+1}^L P_i P_j (m_i - m_j)(m_i - m_j)^T$$

The optimal features are determined by optimizing the Fisher criteria given by

$$J_{DAFE} = \text{tr}(S_w^{-1} S_b Y) \quad , \quad \text{where } Y = A^T X \quad (1)$$

Approximated pairwise accuracy criterion Linear Dimension Reduction (aPAC-LDR) [3] can be seen as DAFE weighted contributions of individual class pairs according to the Euclidian distance of respective class means. The major difference between DAFE and aPAC-LDR is that the Fisher criteria is redefined as

$$J_{a-PAC}(A) = \sum_{i=1}^{L-1} \sum_{j=i+1}^L P_i P_j \omega(\Delta_{ij}) \text{tr}[(AS_w A^T)^{-1} (AS_{ij} A^T)]$$

where $S_{ij} = (m_i - m_j)(m_i - m_j)^T$, $\Delta_{ij} = \sqrt{(m_i - m_j)^T S_w^{-1} (m_i - m_j)}$

$$\text{and } \omega(\Delta_{ij}) = \frac{1}{2\Delta_{ij}^2} \text{erf}\left(\frac{\Delta_{ij}}{2\sqrt{2}}\right)$$

The above weighted Fisher criteria is the same as (1) by redefining the between-class scatter matrix as

$$S_b = \sum_{i=1}^{L-1} \sum_{j=i+1}^L P_i P_j \omega(\Delta_{ij}) (m_i - m_j)(m_i - m_j)^T \quad (2)$$

Hence the optimization problem is the same as DAFE.

The advantage of DAFE (or aPAC-LDR) is that it is distribution-free but there are three major disadvantages in DAFE (or aPAC-LDR). One is that it works well only if the distributions of classes are normal-like distributions [1]. When the distributions of classes are nonnormal-like or multi-modal mixture distributions, the performance of DAFE is not satisfactory. The second disadvantage of DAFE is the rank of the within-scatter matrix S_w is number of classes (L) - 1, so generally only L-1 features can be extracted. In real situations, the data distributions are often complicated and not normal-like, therefore only using L-1 features is not sufficient for much real data. The third limitation is that if the within-class covariance is singular, which often occurs in high dimensional problems, DAFE will have a poor performance on classification.

2.2 Nonparametric Feature Extractions

Nonparametric Discriminant Analysis (NDA) [2][4] was proposed to solve the problems of DAFE. In NDA, the between-class scatter matrix is redefined as a new nonparametric between-class scatter matrix (for the 2 classes problem), S_b denoted, as

$$S_b = \sum_{i=1}^L P_i \sum_{j \neq i} \sum_{l=1}^{n_i} \frac{w_l^{(i,j)}}{n_i} (x_l^{(i)} - M_j(x_l^{(i)}))(x_l^{(i)} - M_j(x_l^{(i)}))^T$$

where $M_j(x_l^{(i)}) = \frac{1}{k} \sum_{l=1}^k x_{jNN}^{(i)}$ is called the local kNN mean of $x_l^{(i)}$, $x_{jNN}^{(i)}$ is the jth nearest neighborhood (NN) from class i (ω) to the sample x_l , and $x^{(i)}$ refers to samples from class i . The parametric S_w was still suggested to be used in NDA by the authors.

The disadvantages of NDA are: 1. Parameters k and α are usually decided by rules of thumb. So the better result usually comes after several trails. 2. S_w is still with a parametric form. When the training set size is small, NDA

will have the singularity problem.

Nonparametric weighted feature extraction (NWFE) [5] is proposed for improving DAFE and NDA. In NWFE, the nonparametric between-class scatter matrix for L classes is defined as

$$S_b = \sum_{i=1}^L P_i \sum_{j \neq i} \sum_{k=1}^{n_i} \frac{\lambda_k^{(i,j)}}{n_i} (x_k^{(i)} - M_j(x_k^{(i)}))(x_k^{(i)} - M_j(x_k^{(i)}))^T \quad (3)$$

where $x_k^{(i)}$ refers to the k-th sample from class i . Basically, (3) is similar to (2). The differences are in the definitions of weights and local means. The scatter matrix weight $\lambda_k^{(i,j)}$ is a function of $x_k^{(i)}$ and $M_j(x_k^{(i)})$, and defined as:

$$\lambda_k^{(i,j)} = \frac{\text{dist}(x_k^{(i)}, M_j(x_k^{(i)}))^{-1}}{\sum_{l=1}^{n_i} \text{dist}(x_l^{(i)}, M_j(x_l^{(i)}))^{-1}}$$

where $\text{dist}(a,b)$ means the distance from a to b . If the distance between $x_k^{(i)}$ and $M_j(x_k^{(i)})$ is small then its weight $\lambda_k^{(i,j)}$ will be close to 1; otherwise, $\lambda_k^{(i,j)}$ will be close to 0 and sum of total $\lambda_k^{(i,j)}$ for class i is 1. $M_j(x_k^{(i)})$ is the local mean of $x_k^{(i)}$ in the class j and defined as:

$$M_j(x_k^{(i)}) = \sum_{l=1}^{n_j} w_{kl}^{(i,j)} x_l^{(j)}$$

$$\text{where } w_{kl}^{(i,j)} = \frac{\text{dist}(x_k^{(i)}, x_l^{(j)})^{-1}}{\sum_{l=1}^{n_j} \text{dist}(x_k^{(i)}, x_l^{(j)})^{-1}}$$

The weight $w_{kl}^{(i,j)}$ for computing local means is a function of $x_k^{(i)}$ and $x_l^{(j)}$. If the distance between $x_k^{(i)}$ and $x_l^{(j)}$ is small then its weight $w_{kl}^{(i,j)}$ will be close to 1; otherwise, $w_{kl}^{(i,j)}$ will be close to 0 and sum of total $w_{kl}^{(i,j)}$ for $M_j(x_k^{(i)})$ is 1. The nonparametric within-class scatter matrix is defined as

$$S_w = \sum_{i=1}^L P_i \sum_{k=1}^{n_i} \frac{\lambda_k^{(i,i)}}{n_i} (x_k^{(i)} - M_i(x_k^{(i)}))(x_k^{(i)} - M_i(x_k^{(i)}))^T$$

To reduce the effect of the cross products of between-class distances and prevent the singularity, we will regularize by

$$S_w = 0.5 S_w + 0.5 \text{diag}(S_w)$$

The optimal features are determined by optimizing the criteria given by

$$J_{NWFE} = \text{tr}(S_w^{-1} S_b)$$

Finally the NWFE algorithm is

1. Compute the distances between each pair of sample points and form the distance matrix.
2. Compute $w_l^{(i,j)}$ using the distance matrix
3. Use $w_l^{(i,j)}$ to compute local means $M_j(x_k^{(i)})$
4. Compute scatter matrix weight $\lambda_k^{(i,j)}$
5. Compute S_b and S_w
6. Select the m eigenvectors of $S_w^{-1} S_b$, $\psi_1, \psi_2, \dots, \psi_m$, which correspond to the m largest eigenvalues to form the transformation matrix $A_m = [\psi_1, \psi_2, \dots, \psi_m]$

Another feature extraction method included in the comparison of this study is called Generalized Discriminant Analysis (GDA). GDA deals with nonlinear discriminant analysis using kernel function operator. The underlying

theory is close to the Support Vector Machines (SVM) insofar as the GDA method provides a mapping of the input vectors into high dimensional feature space. In the transformed space, linear properties make it easy to extend and generalize the classical Linear Discriminant Analysis (LDA) to nonlinear discriminant analysis. For detail information, see [6].

3. EXPERIMENT DESIGN

In this study, Gaussian, Parzen, and 2NN classifiers are used for comparing the performances of five different feature extraction algorithms, DAFE, aPAC-LDR, NDA(2NN), GDA, and NWFE.

There are three different real data sets, Cuprite, a site of geologic interest in western Nevada, Jasper Ridge, a site of ecological interest in California, and Indian Pine, a mixed forest/agricultural site in Indiana. They were gathered by a sensor known as AVIRIS, mounted in an aircraft flown at 65,000 ft. altitude and operated by the NASA/Jet Propulsion Lab. It produces pixels in 220 spectral bands measuring approximately 20 m across on the ground. Only 191 informative bands are used in experiments. There are 8, 6, and 6 classes in Cuprite, Jasper Ridge, and Indian Pine data sets respectively. There are 40 training samples in each class of Cuprite, Jasper Ridge, and Indian Pine experiments.

4. EXPERIMENT RESULTS

All experiment results are displayed in Figure 2 to 4. We have following findings:

1. In all experiments, using NWFE applied to the Gaussian classifier has the best performance.
2. Figure 2(c) shows that if only 5 (L-1) features are used then the accuracy (expressed as a percentage of the test samples correctly classified) of DAFE is 57% and that of NWFE is 86%. But if 7 features of NWFE are used then the accuracy increases to 91%. This shows that only using L-1 features is not enough in this real situation. DAFE cannot do this due to the restriction of the rank of the between-class scatter matrix. NWFE does not have this restriction.
3. With Gaussian and 2NN classifier, GDA got very poor performances. Only with Parzen classifier, the performance of GDA is improved, but still worse than the combination of NWFE and Gaussian classifier.

5. CONCLUSIONS

The volume available in high dimensional feature spaces is very large, making possible the discrimination between classes with only very subtle differences. On the other hand, this large volume makes increasingly challenging the problem of defining *adequate precisely* the desired classes in terms of the feature space variables. The problems of class statistics estimation error resulting from training sets of finite size grows rapidly with dimensionality, thus mak-

ing it desirable to use no larger feature space dimensionality than necessary for the problem at hand, and therefore the importance of an effective, case-specific feature extraction procedure. The NWFE algorithm presented here is intended to take advantage of the desirable characteristics of DAFE, and NDA, while avoiding their shortcomings. DAFE is fast and easy to apply, but its limitation of L-1 features, its reduced performance particularly when the difference in mean values of classes is small, and the fact that it is based on the statistical description of the entire training set, making it sensitive to outliers, limit its performance in many cases. NDA does not have these limitations. They focus the attention on training samples near the needed decision boundary. NDA does not perform well on unequal covariance or complexly distributed data. NWFE does not have any of these limitations. It appears to have improved performance in a broad set of circumstances, making possible substantially better classification accuracy in the data sets tested, which included sets of agricultural, geological, ecological and urban significance. This improved performance is perhaps due to the fact that, like NDA, attention is focused upon training samples that are near to the eventual decision boundary, rather than equally weighted on all training pixels as with DAFE. It also appears to provide feature sets which are relatively insensitive to the precise choice of feature set size, since the accuracy versus dimensionality curves are relatively flat beyond the initial knee of the curve. This characteristic would appear to be significant for the circumstance when this technology begins to be used by general remote sensing practitioners who are not otherwise highly versed in signal processing principles and thus might not realize how to choose the right dimensionality to use.

REFERENCES

- [1] D. A. Landgrebe, "Information Extraction Principles and Methods for Multispectral and Hyperspectral Image Data," Chapter 1 of *Information Processing for Remote Sensing*, edited by C. H. Chen, published by the World Scientific Publishing Co., Inc., 1060 Main Street, River Edge, NJ 07661, USA 1999.
- [2] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, San Diego: Academic Press Inc., 1990.
- [3] R. P. W. Duin and R. Haeb-Umbach, "Multiclass Linear Dimension Reduction by Weighted Pairwise Fisher Criteria," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, 2001, pp. 762-766.
- [4] K. Fukunaga and M. Mantock, Nonparametric Discriminant Analysis, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 5, 1983, pp. 671-678.
- [5] B-C. Kuo and D. A. Landgrebe, Improved Statistics Estimation And Feature Extraction For Hyperspectral Data Classification, *PhD Thesis and School of Electrical & Computer Engineering Technical Report*. TR-ECE 01-6, December 2001.
- [6] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12:2385-2404, 2000
- [7] G. F. Hughes, "On the mean accuracy of statistical pattern recognition", *IEEE Trans. Information Theory*, 1968, vol. IT-14, no. 1, pp 55-63

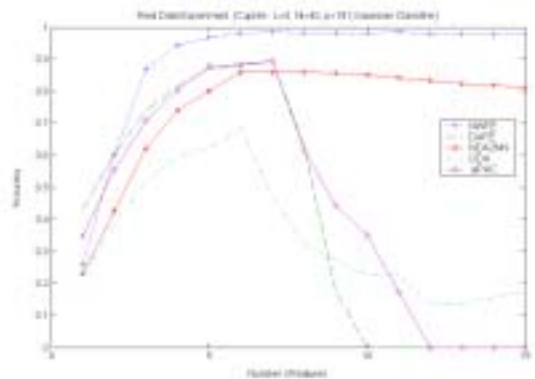


Figure 2(a) Cuprite: Mean of accuracies of Gaussian classifier using 1~15 features

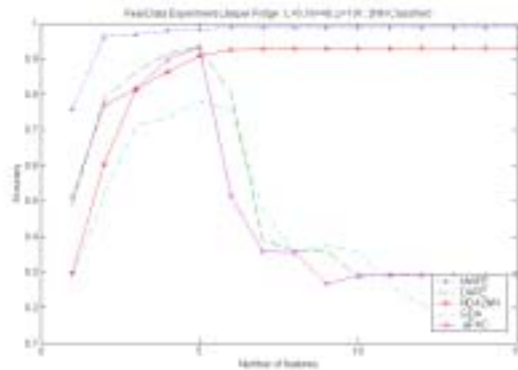


Figure 3(b) Jasper Ridge: Mean of accuracies of Gaussian classifier using 1~15 features

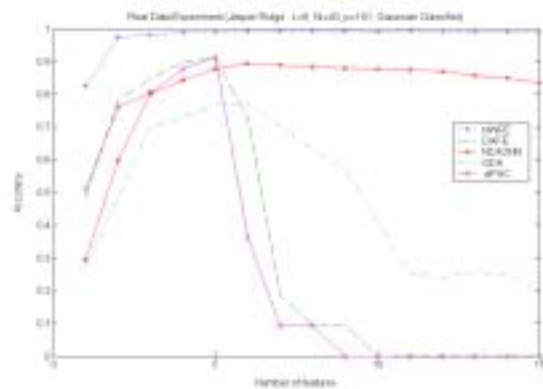


Figure 2(b) Jasper Ridge: Mean of accuracies of Gaussian classifier using 1~15 features

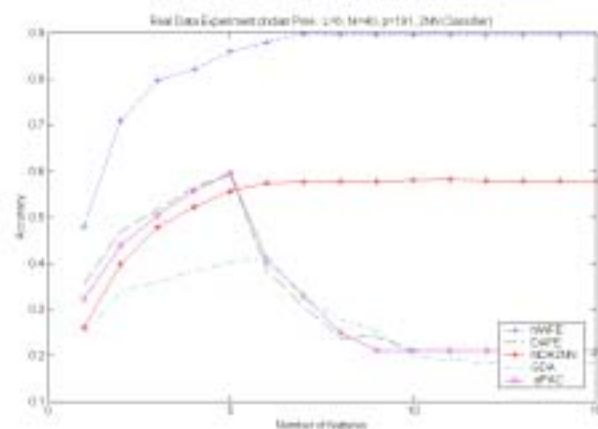


Figure 3(c) Indian Pine: Mean of accuracies of 2NN classifier using 1~15 features

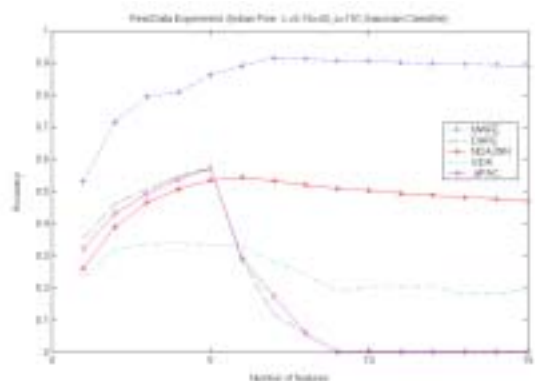


Figure 2(c) Indian Pine: Mean of accuracies of Gaussian classifier using 1~15 features

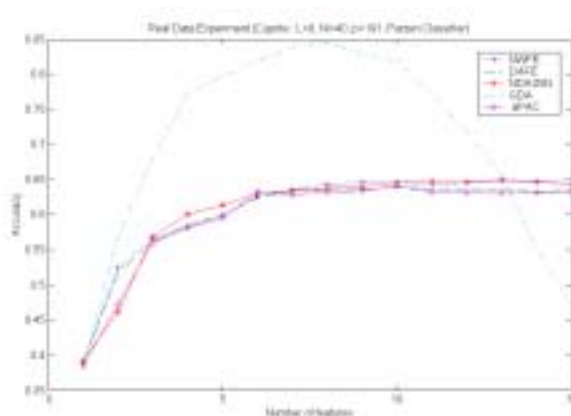


Figure 4(a) Cuprite: mean of accuracies of Parzen classifier using 1~15 features

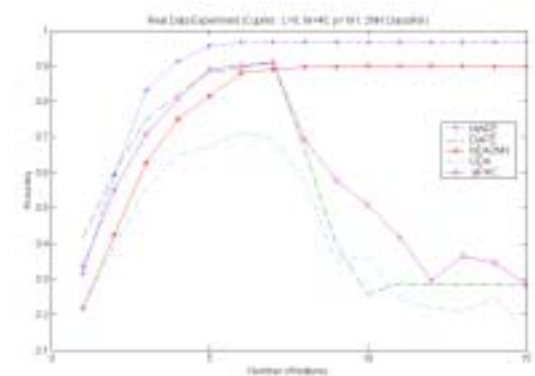


Figure 3(a) Cuprite: Mean of accuracies of 2NN classifier using 1~15 features

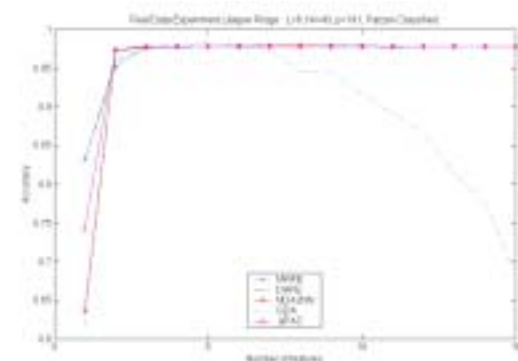


Figure 4(b) Jasper Ridge: Mean of accuracies of Parzen classifier using 1~15 features

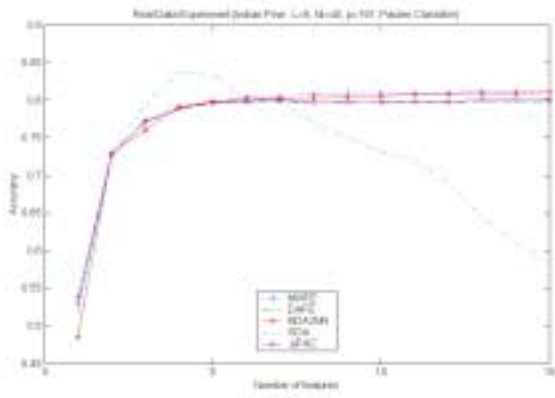


Figure 4(c) Indian Pine: Mean of accuracies of Parzen classifier using 1~15 features