Linear Feature Extractors Based on Mutual Information

Kurt D. Bollacker and Joydeep Ghosh
Department of Electrical and Computer Engineering,
University of Texas, Austin, Texas 78712, USA
kdb@pine.ece.utexas.edu, *ghosh@pine.ece.utexas.edu*

Abstract

This paper presents and evaluates two linear feature extractors based on mutual information. These feature extractors consider general dependencies between features and class labels, as opposed to well known linear methods such as PCA which does not consider class labels and LDA, which uses only simple low order dependencies. As evidenced by several simulations on high dimensional data sets, the proposed techniques provide superior feature extraction and better dimensionality reduction while having similar computational requirements.

1. Introduction

The capabilities of a classifier are ultimately limited by the quality of the features in each input vector. In particular, when the measurement space is high-dimensional but the number of samples is limited, one is faced with the "curse of dimensionality" problem during training [3]. Feature extraction is often used to alleviate this problem. Although linear feature extractors are ultimately less flexible than the more general non-linear extractors, they have some useful properties that can make them highly desirable. Linear projections tend to be structure preserving and have only small, predictable computational demands. Also, under certain conditions linear transforms preserve all useful information in the original feature set.

One of the most commonly used unsupervised linear feature extractors, the Karhunen-Loève Expansion, performs principal component analysis (PCA) using covariance between original features as an extraction criterion. A leading *supervised* linear feature extractor is linear discriminant analysis (LDA), a generalization for c classes of Fisher's linear discriminant [7][11], where the extraction criterion used is class mean separation.

In this paper, we present two supervised linear feature extractors which use mutual information as a feature extraction criterion. The performance of these feature extractors is compared empirically with PCA, LDA, and a mutual information feature selector, using four separate classification problems. The results are summarized along with a discussion of computational complexity.

2. Mutual Information Feature Extraction Criteria

One definition of an optimal mutual information feature extractor $f^*(\cdot)$ is

$$f^*(\vec{X}) = \max_{f \in \mathcal{F}} I(f(\vec{X}); Y) \qquad (1)$$

where X is the input vector, Y is the output vector, and $\mathcal{F}$ is the space of all considered feature extractors. If $\mathcal{F}$ is the space of all linear transforms, then this equation becomes

$$\mathbf{A}^*\vec{X} = \max_{\mathbf{A} \in \mathcal{F}} I(\mathbf{A}\vec{X}; Y) \qquad (2)$$

If the $\mathbf{A}$ matrix is $n \times n$ and simply rotates or flips the coordinate system without scaling, then the linear transform will not destroy information. More exactly, for an n dimensional input vector if A is chosen to be a real, $n \times n$, non-singular matrix with $\|\mathbf{A}\vec{X}\| = \|\vec{X}\|$ $\forall \vec{X}$, then it is true that

$$I(\vec{X}, Y) = I(\mathbf{A}\vec{X}; Y) \qquad (3)$$

A good feature extractor would be one that allows for $m : m < n$ of the extracted features to be chosen while minimizing the information loss. One of several standard non-linear optimization techniques [6] could be used to directly solve for an optimal $m \times n$ matrix $\mathbf{A}^*$ directly from the distributions of $\vec{X}$ and Y. However, the density of p points in an n dimensional input space is proportional to $p^{(\frac{1}{n})}$, making the sampling density of $\mathbf{A}\vec{X}$ in high dimensional space very low. The resulting poor confidence for a numerical approximation of $I(\mathbf{A}\vec{X};Y)$ suggests that an alternate approach must be used.

Comparison of Feature Extraction Criteria: PCA uses covariance as a feature extractor criterion, This relation to correlation only measures the linear dependence among features. LDA uses the simple first order statistic of distance of feature vector means criterion. Mutual information, measures a general dependence of class labels and the extracted features [5] and can be used to measure this dependence when the class labels are unordered. Thus, a mutual information measure can be a more powerful feature extractor criterion.

Previous MI based Feature Selectors: Battiti has developed a mutual information feature selection (MIFS) algorithm using input feature distributions and the class distribution [1]. In this model, features and the class labels are treated as (sampled) random variables. The i_{th} best feature f_i selected in the MIFS algorithm is that which satisfies $f_i = \max_{X_i} I(Y;X_i) - \beta \sum_{j=1}^{i-1} I(X_i;X_j)$ where Y is the class label random variable, X_i is the i_{th} input feature random variable and β is a tunable paramenter. This criterion greedily selects the set of features with high mutual information with the class labels while trying to minimize the mutual information among chosen features.

Also, Sheinvald, Dom and Niblack developed a feature selection algorithm based on Minimum Description Length, a criterion related to mutual information [10]. However, like MIFS, this is only a feature selector, and thus is less powerful than a general feature extractor.

3. Mutual Information Feature Extractors

This section presents two feature extractors which are designed to maximize the mutual information of the extracted features with the output. These feature extractors do not require a mutual information calculation of an n dimensional feature vector and both satisfy the requirements such that Equation 3 holds.

Maximum Mutual Information Projection Feature Extractor(MMIP): This feature extractor attempts to find successive orthogonal normalized projections of the input vector which maximize mutual information with the output distribution. The first such projection $\vec{a}^*$ is defined by

$$\vec{a}^* = \max_{\vec{a}} I(\vec{a}^T \vec{X};Y) \tag{4}$$

where $\vec{a}^{*T}\vec{X}$ is the extracted feature which has the highest mutual information with the class label Y. The mutual information was numerically approximated using the method described by Battiti [1] with 25 equal sized intervals. The polytope algorithm [4] was used to find an approximately maximal projection. This projection is then removed from the input vectors, and the maximal mutual information projection of the residuals is found again $m-1$ times in this manner. It should be noted that this feature extraction method suffers from the problem of overlapping mutual information contributed by each feature, which becomes worse as more features are extracted.

A Separated Mutual Information Feature Extractor(SMIFE): A heuristic similar in form to PCA attempts to extract features which have high mutual information with the output. In PCA, the eigenvectors of large eigenvalues are uncorrelated projections [2] in which variance is high. In place of covariance values in the matrix, three-variable mutual information values are used. Three-variable mutual information is defined as

$$I(X_i;X_j;Y) = H(X_i,X_j,Y) - H(X_i) - H(X_j) - H(Y) + I(X_i;Y) + I(X_j;Y) + I(X_i;X_j) \tag{5}$$

where X_i and X_j are features, Y is the class label and $H(\cdot)$ is the entropy function. The eigenvectors of this mutual information matrix are found and ordered by decreasing eigenvalue. Following an analogy with PCA, the principal components should be directions of high mutual information with the class label and should minimize common mutual information with the class label. As with PCA, Equation 3 holds for this feature extractor. Two versions, SMIFE1 (above) and SMIFE2 were constructed. For SMIFE2, the terms of Equation 5 are rewritten in the form

$$I(X_i;X_j;Y) = I(X_i;Y) + I(X_j;Y) - I(X_i,X_j;Y) \tag{6}$$

Instead of finding eigenvectors of a matrix of $I(X_i;X_j;Y)$ values ordered by decreasing eigenvalue, the eigenvectors of a matrix of $I(X_i,X_j;Y)$ values were found and due to the negative sign, ordered by increasing eigenvalue.

4. Experiments

Methods: Four different classification problem data sets from the UC Irvine database were used for experimentation with the feature extractors (Table 1). The chosen data sets were: letter recognition (LR), a segmented image classifier (SEG), a satellite image classifier (SAT), and a vehicle silhouette classifier (VEH).

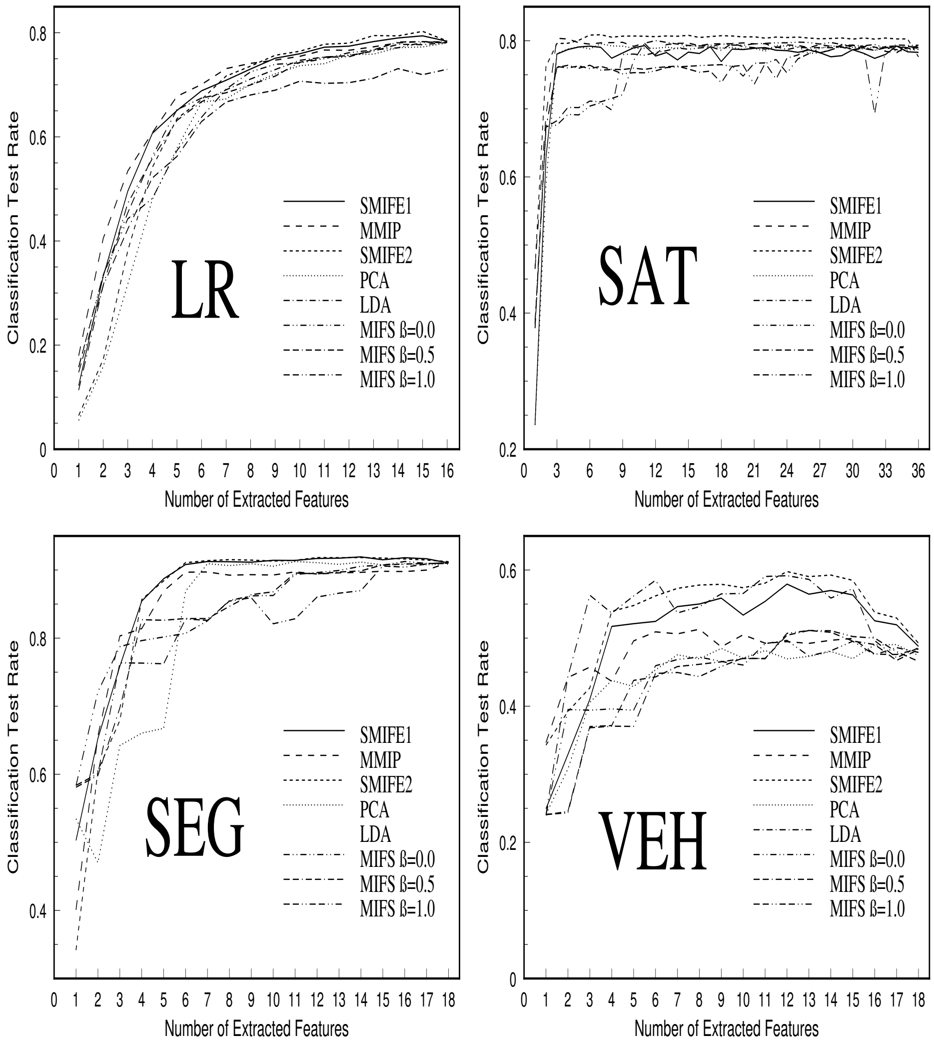

Figure 1. Test Rate versus number of features for the four data sets.(LDA is not included for the SEG data set because the between class scatter matrix was almost singular.)

The examples were randomly divided into evenly sized training and test data sets. A fully connected back-propagation multilayer perceptron (MLP) network was trained for each data set and an attempt was made to refine the architectural and learning parameters to optimize the test performance on the original feature set. Several feature extractors were used on the four different data sets. PCA, LDA and three variations of Battiti's feature selector ($\beta = 0.0$,$\beta = 0.5$,$\beta = 1.0$) as well as the MMIP, SMIFE1, and SMIFE2 feature extractors were compared. The LDA considered was Okada and Tomita's[8] extension which does not share traditional LDA limit on the number of features extracted. For each feature extractor and data set with n original features, n MLP's were trained with the m best extracted features where $m = 1 \ldots n$. The classification test rate was the measure of performance used. For three of the data sets test results from 10 runs of each MLP were

Table 1. The data sets used for feature extractor comparison

Data Set	# Features	# Classes	# Examples
LR	16	26	20000
SEG	18	7	2310
SAT	36	6	6435
VEH	18	4	846

Table 3. The required number of features (#f) required for best classification performance and the associated testing rates

Feature Extractor	Data Set							
	LR		SEG		SAT		VEH	
	#f	Test Rate	#f	Test Rate	#f	Test Rate	#f	Test Rate
MMIP	16	0.7820	18	0.9110	3	0.8036	8	0.5130
SMIFE1	15	0.7940	14	0.9193	34	0.7923	12	0.5793
SMIFE2	15	0.8026	12	0.9182	7	0.8088	12	0.5977
PCA	16	0.7804	11	0.9124	17	0.7938	17	0.4904
LDA	16	0.7814	N/A	N/A	12	0.8002	12	0.5917
MIFS $\beta = 0.0$	16	0.7832	16	0.9125	25	0.7979	12	0.4966
MIFS $\beta = 0.5$	14	0.7310	17	0.9096	29	0.7942	13	0.5111
MIFS $\beta = 1.0$	15	0.7828	18	0.9108	34	0.7921	13	0.5111

averaged to compensate for performance variance while 50 runs were used in the VEH data set due to its higher variance.

Table 2. The best feature extractor for each data set using the m best extracted features

m	Data Set			
	LR	SEG	SAT	VEH
1	MMIP	MIFS1.0	SMIFE2	MMIP
2	MMIP	MIFS0.0	SMIFE2	LDA
3	MMIP	MMIP	MMIP	LDA
4	MMIP	SMIFE2	MMIP	SMIFE2
5	MMIP	SMIFE1	SMIFE2	LDA
6	MMIP	SMIFE2	SMIFE2	LDA
7-8	MMIP	SMIFE2	SMIFE2	SMIFE2
9	SMIFE2	SMIFE2	SMIFE2	SMIFE2
10	SMIFE2	SMIFE1	SMIFE2	SMIFE2
11	SMIFE2	SMIFE2	SMIFE2	LDA
12-13	SMIFE2	SMIFE2	SMIFE2	SMIFE2
14	SMIFE2	SMIFE1	SMIFE2	SMIFE2
15	SMIFE2	SMIFE2	SMIFE2	SMIFE2
16	SMIFE2	SMIFE1	SMIFE2	SMIFE2
17		SMIFE1	SMIFE2	SMIFE2
18		SMIFE2	SMIFE2	SMIFE2
19-35			SMIFE2	
36			MIFS0.5	

Results: Two measures of test performance were made from the results shown in figure 1. First, for each quantity of extracted inputs the best performing feature extractor was listed. This is compiled in Table 2. Second, the maximum performance and the number of features required for that performance were measured for each feature extractor as shown in Table 3.

Table 4. Computational complexity of each feature extractor (n_s = Number of samples, n = Original feature dimensionality)

Feature Extractor	Computational Complexity
SMIFE1	$O(n_s \times n^2)$
SMIFE2	$O(n_s \times n^2)$
MMIP	$O(n_s \times n \times Avg\#iterations)$
PCA	$O(n_s \times n^2)$
LDA	$MAX(O(n^4), O(n_s \times n^2))$
MIFS	$O(n_s \times n^2)$

Discussion: From Table 2 the best performing feature extractor versus the m best extracted features can be seen. MMIP does best for low numbers of features in the LR data sets while several feature extractors are close in the SAT data set. MIFS does well in the SEG data set and LDA does well in the VEH data set. MMIP's best performance only with a small number of features seems to give evidence that as more features are used, less new information is added for classification because of the information overlap between features. Because the within class matrix is almost singular for the SEG data set, LDA cannot be used. However, LDA performs very well on the VEH data set, being the best performer for small numbers of features. When enough extracted features are included, SMIFE2 performs best on all of the data sets except SEG, where SMIFE1 does just about as well and VEH where LDA is a very close performer. The SMIFEs' performance gives evidence that information separation occurs which allows for better performance when more extracted features are added, despite the increased dimensionality.

Table 3 shows the best performance achieved by each feature extractor, and how many features were needed to reach that performance. In the LR, SEG, and SAT data sets, the best performance is very close for all of the feature extractors. However, in the VEH data set both SMIFEs and LDA enjoy a significant performance advantage over the other extractors, but SMIFE2 outperforms LDA slightly.

Computational Complexity of Feature Extraction: One calculation of covariance for PCA had a complexity of $O(n_s)$ where n_s is the number of samples. PCA requires $O(n^2)$ such calculations plus the eigenvector and eigenvalue calculations for a $n \times n$ matrix, typically $O(n^3)$[9]. Okada and Tomita's LDA requires $O(n_s \times n^2)$ calculations to generate the scatter matrices and $O(n^4)$ calculations to find the n optimal projections. The numerical calculation of mutual information for the MIFS and MMIP feature extractors is complexity $MAX(O(n_i \times n_c), O(n_s))$ where n_c is the number of classes and n_i is the number of intervals, chosen to be a constant 25. Dropping n_i out, and given that n_c is almost certainly always much less than n_s, a complexity of $O(n_s)$ is left. MIFS requires $O(n^2)$ such calculations but the iterative nature of MMIP varies the number of requires mutual information calculations. In the four data sets, somewhere between 200 and 700 iterations was typical per feature extracted. Both the calculations of three variable mutual information for SMIFE1 and the joint mutual information for SMIFE 2 require $MAX(O(n_i^2 \times n_c), O(n_s))$, but since n_i is a constant, the complexity is still $O(n_s)$. A summary of this can be seen in Table 4. MIFS, PCA and SMIFE all have the same order of complexity but, but LDA had the potential highest, being the maximum of $O(n^4)$ and $O(n_s \times n^2)$) while MMIP's complexity varies with the data set. However, it should be kept in mind that a better optimization algorithm would likely improve upon the inefficient polytope. Even with polytope, the MMIP algorithm required (as an example) only 2.2 minutes on a CPU capable of 100 SPECfp92 to find all 18 SEG data set features.

5. Conclusions

Mutual information as a feature extraction criterion is used to guide the design of two linear feature extractors. These feature extractors performed better than PCA and Battiti's MIFS feature selectors while having the same order of computational complexity. The performance of SMIFE was better than or about equal to LDA while having the lower computational complexity. Since PCA is an optimal variance extractor, there is thus empirical evidence that mutual information is a better linear feature extractor criterion. LDA has a higher computational complexity and can fail on some data sets. Also, the general feature extractors SMIFE and MMIP performed better than the mutual information based feature selector MIFS. Future work should include a more rigorous argument for the usefulness of the SMIFE feature extractors and a proof of optimality as well as comparison with other linear feature extractors.

Acknowledgements: This research is supported in part by AFOSR grant F49620-93-1-0307 and ARO contracts DAAH 04-94-G0417 and 04-95-10494. K. Bollacker is an NSF Graduate Fellow and is also supported by the the Cockrell Foundation. We also thank Bryan Stiles and Viswanath Ramamurti for their help.

References

[1] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5:537–550, July 1994.

[2] P. A. Devijer and J. Kittler. *Pattern Recognition: A Statistical Approach.* Prentice Hall International, Englewood Cliffs, New Jersey, 1982.

[3] J. H. Friedman. An overview of predictive learning and function approximation. In J. Friedman and W. Stuetzle, editors, *From Statistics to Neural Networks,Proc. NATO/ASI Workshop.* Springer Verlag, 1994.

[4] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization.* Harcourt Brace and Company, London, 1981.

[5] W. Li. Mutual information functions versus correlation functions. *Journal of Statistical Physics*, 60:823–837, 5/6 1990.

[6] D. G. Luenberger. *Linear and Nonlinear Programming.* Addison Wesley, Massachusetts, 1984.

[7] J. Mao and A. Jain. Aritficial neural networks for feature extraction and multivariate data proejction. *IEEE Transactions on Neural Networks*, 6:296–317, 2 1995.

[8] T. Okada and S. Tomita. An optimal orthonormal system for discriminant analysis. *Pattern Recognition*, 18:139–144, 2 1985.

[9] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, editors. *Numerical Recipes in C: The Art of Scientific Computing.* Cambride University Press, Cambridge, 1992.

[10] J. Sheinvald, B. Dom, and W. Niblack. A modeling approach to feature selection. In *10th International Conference on Pattern Recognition*, pages 535–539, june 1990.

[11] W. Siedlecki, K. Siedlecka, and J. Sklansky. An overview of mapping techniques for exploratory pattern analysis. *Pattern Recognition*, 21:411–429, 5 1988.