European Journal of Scientific Research
ISSN 1450-216X Vol.34 No.2 (2009), pp.196-203

http://www.eurojournals.com/ejsr.htm

English Letter Classification Using Bayesian Decision Theory and Feature Extraction Using Principal Component Analysis

Mujtaba Husnain
Department of Computer Science & IT
the Islamia University of Bahawalpur, Pakistan
E-mail: arhusnain@hotmail.com
Tel: 092-062-9255466

Shahid Naweed
Department of Computer Science & IT
the Islamia University of Bahawalpur, Pakistan
E-mail: shahid_naweed@hotmail.com
Tel: +092-062-9255466

Abstract

This paper uses Bayesian decision Theory (BDT), one of the statistical techniques for pattern classification, to identify each of the large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet. The character images were based on 20 different fonts and each letter within 20 fonts was randomly distorted to produce a file of 20,000 unique instances. The features of the dataset and the errors committed by Holland-style adaptive classifiers were analyzed in an attempt to use BDT in-order to reduce the error rate. At the end, Principal Component Analysis (PCA) is applied for dimensionality reduction. Empirical results on character recognition from the UCI dataset repository are presented.

Keywords: Bayesian Decision Theory, Maximum a-posterior probability, Principal Component Analysis

1. Introduction

The letter image recognition data was donated by David J. Slate and P.W. Frey in 1991 to UCI data repository for the researchers and scientists to analyze the patterns in efficient way. In this paper classical statistical technique Bayesian Decision Theory is used to recognize and classify the pattern of English capital alphabet among 26-alphabet classes. Improvement is gained by reducing the error rate down to 2%. Finally, Principal Component Analysis (PCA) is used for feature extraction to reduce the dimensions of the pattern data.

1.1. Dataset

The image dataset comprises of 20,000 instances/images of English capital alphabet. Different distortion techniques are applied on the image data such as compress, change aspect ratio along with x

and y-axis etc; to add some bearable noise. About 20 font types are selected with different stroke styles and about six different letter styles.

1.2. Image Feature Description

For each black-and-white image of the English alphabet, 16-dimensional feature vector was extracted by the author to demonstrate the summary of the alphabet image [1]. This feature vector contains the characteristic features of the image such as vertical and horizontal position of the rectangular box containing the alphabet, total number of *ON* pixels, edge count etc. The full description of the feature vector can be found in [1].

2. Literature Review

The lowest classification error rate of 17% was observed by the Frey and Slate by applying Holland-style adaptive classification [1]. About 16,000 stimuli were observed under training and remaining 4000 were tested for classification. Improvement in accuracy of 14% was obtained after from experimental results of Schapire and Freud who applied Decision Tree algorithm for classification [4]. Later, the empirical results of Schwenk and Bengio reduced the error rate down to 6.2 % by applying fully connected MLP [5]. Further the error rate was reduced to 2.6% by Breuel after using adaptive statistical similarity [3].

3. Research Method

To establish the background, let us review the basic concepts of Bayesian Decision Theory (BDT). It is a fundamental statistical approach to the generic pattern classification problems. It makes the assumption that the solution to pattern classification problem is purely based on probabilistic values and all the relevant probability values are known. The decision rule of BDT says that for minimum error rate classifier, we should choose the class with minimum posterior probability [6].

The explanation is as under: let $\boldsymbol{\lambda}$ be finite set of classes $\boldsymbol{c_1, c_2, c_3 \ldots c_n}$ and our unknown feature vector $\boldsymbol{x}$, where $\boldsymbol{x}$ ($\in \mathbb{R}$) is a *d*-dimensional vector. After calculating conditional posterior probabilities of every class of $\boldsymbol{\lambda}$, choose the class $\mathbf{c} \in \boldsymbol{\lambda}$ for which the posteriori is maximum. The estimation of $P(Ci \mid x)$ depends on estimated value of $P(x \mid Ci)$, the likelihood. The generic Bayes Rule is given by

$$P(C_n \mid x) = \frac{P(x \mid C_n)P(C_n)}{P(x)} \tag{1}$$

Where $P(C)$ is prior probability and $P(x)$ is marginal probability, aka evidence [9]. Where $P(x)$ is calculated by

$$P(x) = \sum_{n=1}^{m} p(x / C_n)P(c_n) \tag{2}$$

Eq. 1 can be written as

$$posterior = \frac{likelihood \times prior}{evidence} \tag{3}$$

The main problem in pattern classification problems is to calculate the conditional probability density values $P(x \mid C_n)$ which are unknown. As feature vector $\boldsymbol{x}$ is generated by a per-class prototype $\boldsymbol{x_i}$ where the dataset is in bulk, the distribution of the data is assumed to be Normal (Gaussian). This assumption plays a vital role in correct prediction of pattern classification [4, 9]. The value

$P(x|C)$ can be predicted by other techniques like logistic regression but this is not the objective of this paper. Our target so far is to find the class with maximum posterior probability $\max_{\lambda} P(C|x)$ with minimum error rate, not just $P(x|C)$. The statistical decision theory can be formulated resorting to the Bayes theory introducing the concept of a risk defined as the expected value of the error cost function. If the latter is assumed to be either a quadratic function or a uniform function, then the Maximum A posteriori Probability (MAP) inference solutions can be calculated.

4. Character Recognition

The above concepts are applied experimentally on a dataset consisting of raw data of black-and-white rectangular pixel that displays one of 26 capital-letters of English alphabets. Each instance was converted into 16 primitive numerical attributes (mean, variance, moments, covariance) scaled to fit into a range of integer value from 0 to 15. The detail of 16 attributes is given below. [1]

1. letter: capital letter (26 values from A to Z)
2. x-box: horizontal position of box (integer)
3. y-box: vertical position of box (integer)
4. width: width of box (integer)
5. height: height of box (integer)
6. onpix: total # on pixels (integer)
7. x-bar: mean x of on pixels in box (integer)
8. y-bar: mean y of on pixels in box (integer)
9. x2bar: mean x variance (integer)
10. y2bar: mean y variance (integer)
11. xybar: mean x y correlation (integer)
12. x2ybr: mean of x * x * y (integer)
13. xy2br: mean of x * y * y (integer)
14. x-edge: mean edge count left to right (integer)
15. xegvy: correlation of x-edge with y (integer)
16. y-edge: mean edge count bottom to top (integer)
17. yegvx: correlation of y-edge with x (integer)

Missing Attribute Values: None

Class Distribution

789 A	766 B	736 C	805 D	768 E	775 F	773 G
734 H	755 I	747 J	739 K	761 L	792 M	783 N
753 O	803 P	783 Q	758 R	748 S	796 T	813 U
764 V	752 W	787 X	786 Y	734 Z		

At the initial stage, about 14000 items were trained and remaining 6000 were predicted one by one for their corresponding alphabet class.

In the experiment, one instance from the testing data set is selected at random and plugged into classifier to check its corresponding alphabet class. The accuracy of the classifier is checked by introducing random 100 input instances it was measured about 92% correct i.e. 8 out of 100 input stimuli are misclassified.

In second set of experiment, the training data is increased from 14000 to 16000 in order to reduce error rate and 100 random character data are selected and plugged, one by one, into the classifier. This resulted in refinement of error rate which was reduced to 2% i.e. only 2 out of 100 stimuli were misclassified. These results are far better than the Holland-Style Adaptive classifier which has the accuracy of only 80%. In this classifier fuzzy logic is used that is based on IF-THEN structure. The bigger disadvantage of the classifier is it takes much more time in classifying the new instance.

The rule of thumb is, among different classifiers choose the classifier having better response time even have less accuracy than the classifier having not better response time even having high accuracy [2, 9].

The above results were also better than the experimental results of Thomas M. Breuel and found 98% accurate as compared with the results of Breuel that were 5.1%. Some of the important experimental results are shown in Fig. 1. The summary of the error rate of comparative results is shown below in Table 1.

Table 1: The error rate comparison of Statistical Adaptive Similarity using nearest neighbor, Holland Style Adaptive Classification and Bayesian Decision Theory (BDT) with dataset of 14000 training samples out of 20,000 total samples.

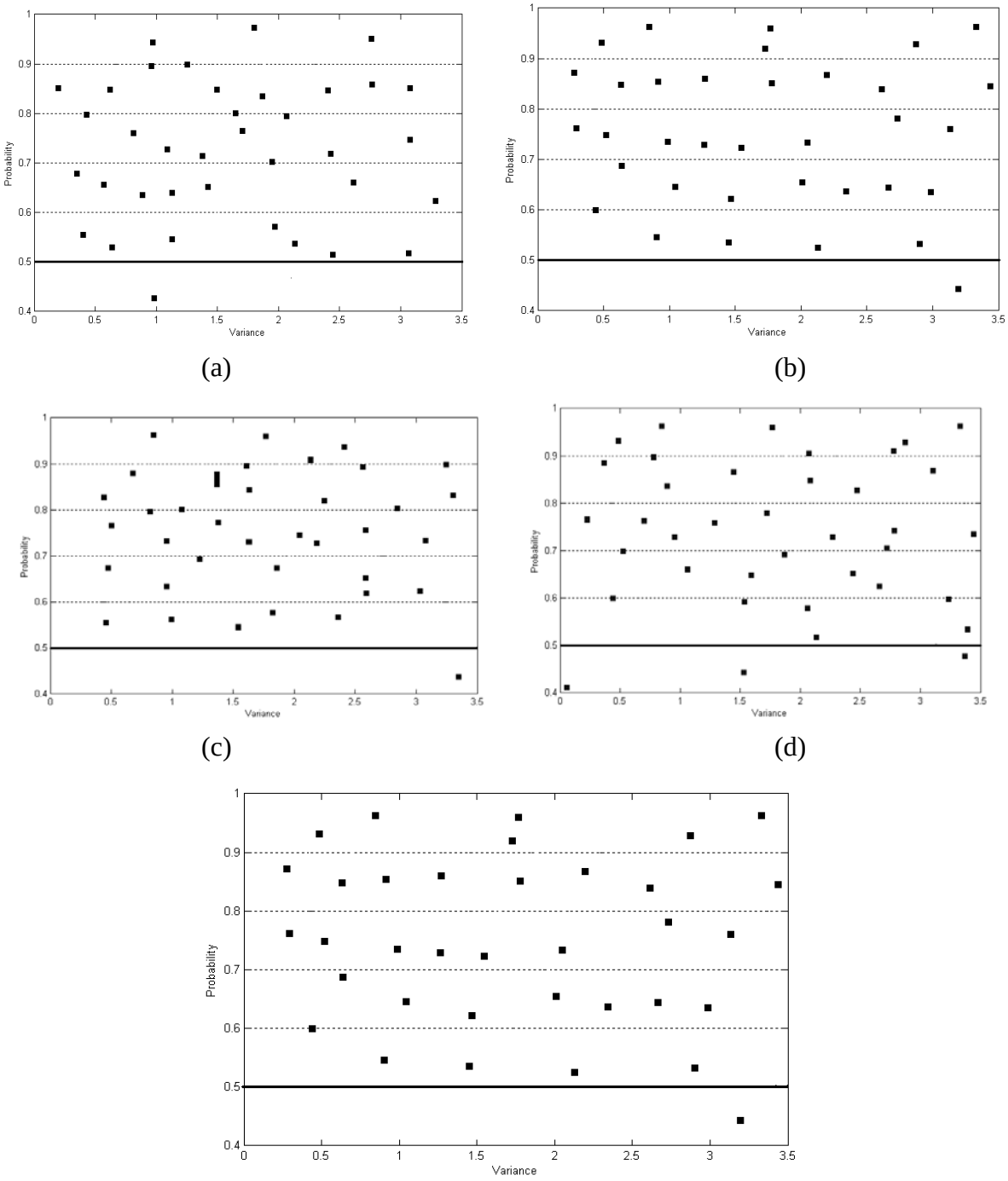
Statistical Adaptive Similarity [4]	Holland_Style Adaptive Classification [1]	Bayesian Classification (our proposed model)
5.1%	17.3%	2%

The most confusing letters in English alphabets are *N* and *H* as both of the letters have almost same shape and same number of *ON* pixels in the character image so there is need of having strong apriori knowledge of the two letters. The posterior probability graphs of the two letters after applying BDT are almost same (Fig. 1 (c) and (d)). The conventional method of classifying the patterns by measuring the nearest neighbor (Euclidean distance) did not give the appropriate results [4]. Fig.1 (e) shows the posterior probability graph of letter *S* which has so many font styles than any other letter. The thick black line in the Fig. 1 is the threshold value of the posterior probability below which the recognized letter will be considered as misclassified. This rule of thumb is algebraically described in equation 2 [4, 7, 9].

$$\begin{aligned} &choose P(C_n \mid x) \\ &P(C_n \mid x) >= 0.5 \\ &OR \\ &P(C_n \mid x) > P(C_m \mid x) \forall n \neq m \end{aligned} \tag{4}$$

The empirical results for posteriori are given in Figure 1 below.

Figure 1: Posteriori of the occurrences of random inputs of **(a)** alphabet A **(b)** alphabet B **(c)** alphabet H **(d)** alphabet N **(e)** alphabet S. Each black dot in each graph represents one input instance. Only first forty inputs of above alphabets are shown to reduce ink-noise ratio.



5. Dimensionality Reduction using PCA

Principal Component Analysis is an eigenvector/value-based approach used in dimensionality reduction (or feature extraction) of the multivariate data. It is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. Since patterns in data can be hard to find in data of high dimension, where the luxury of graphical representation is not available, PCA is a powerful tool for analyzing data. The other main advantage of PCA is that once we

have found these patterns in the data, we compress the data, i.e. by reducing the number of dimensions, without much loss of information. It is widely used in most of the pattern recognition applications like face recognition, image compression, and is common technique for finding patterns in high dimensional data [2].

The question arises why PCA is used for dimensionality reduction even if there are many other techniques are available. The answer lies in the following table, which shows the comparison of different techniques with their pros and cons [2]. Table 2 lists most of the well-known feature selection methods. Only the first two methods in this table guarantee an optimal subset. All other strategies are sub optimal due the fact that the best pair of features need not contain the best single feature. In general, good, larger feature sets do not necessarily include the good, small sets. As a result, the simple method of selecting just the best individual features may fail dramatically.

While applying PCA, we are interested in search of solution of two key problems. 1) To how many dimensions the data should be reduced? 2) How much sample data will be required to estimate the required number of dimensions of pattern data? The answer to first problem lies in analysis of eigenvalues of sample covariance matrix.

In general, once eigenvectors are found from the covariance matrix, the next step is to order them by eigenvalue, highest to lowest. This gives us the components in order of significance. Now, if we like, we can decide to ignore the components of lesser significance.

Table 2: Feature Extraction Methods

Method	**Property**
Principal Component Analysis	Fast, Eigenvector based
Linear Discriminant Analysis	Supervised linear map; fast; eigenvector based
Projection Pursuit	Linear map, iterative, non-Gaussian
Independent Component Analysis	Nonlinear map; eigenvector based
PCA Network	Linear map; non-Gaussian; iterative
Nonlinear auto associative Network	Nonlinear map; iterative; non-Gaussian criterion
Multidimensional Scaling (MDS)	Nonlinear map; iterative

The eigenvalues of the multivariate English alphabet data with 16-attributes are depicted below:

Column 1 through 16:

24.5194	12.8843	10.6937	7.4828	6.4996	4.7999	4.3405	3.3585	2.6937	2.0233
1.5006	1.3737	1.2754	1.0598	0.6872					
0.3118	0								

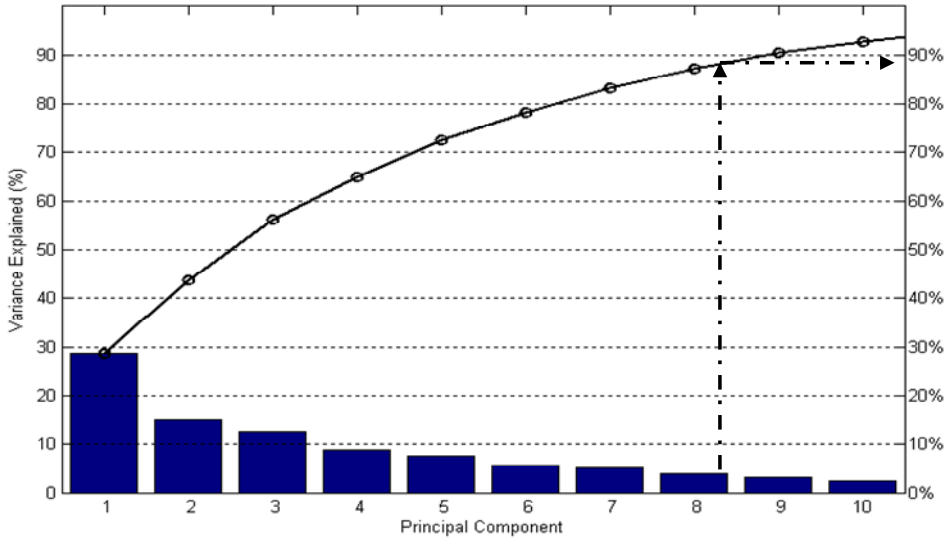
These values will help in selecting the principal components. The rule of thumb is to develop a *threshold value* to choose k principal components (η) out of d dimensional data (where $k < d$). The *threshold* value can be computed by the formula given below [7, 9].

$$\sum_{i=1}^{k} \eta_i \Big/ \sum_{i=1}^{d} \eta_i \rangle 0.9 \text{ (}\textit{threshold value}\text{)} \quad (5)$$

By using above formula first eight principal components were selected. Hence the dimensionality of whole data is reduced from 16 to 8. This result can also be deduced by scree graph drawn in Figure 2. It is obvious from the graph that first eight principal components are showing about 90% of the variance. Adding other eigenvector, at the point indicated by the arrow in figure, will not increase the variance explained.

With the third set of experiments, the efficiency of BDT classifier is checked again for 100 random inputs with the data having reduced dimensions of 8 instead of 16. The accuracy was measured up to 98% correct with 16000 instances kept as training data. PCA, actually, projects the original data along the directions where the data varies the most. These directions are determined by eigenvectors of covariance matrix corresponding to the largest eigenvalue. The magnitude of the eigenvalues corresponds to the variance of the data along the eigenvalue directions [3, 8, 9]. The dimensions are in the data are highly correlated, so the eigenvalues are not small resulting large reduction in dimensionality.

Figure 2: Screegraph showing proportion of the variance explained against principal components. The first 8 eigenvectors (principal components) showing very near to 90% of the variance. Remaining components have no much significance in classifying and their variance values are also diminished (shown in bars).



6. Conclusions & Future Work

This paper has introduced the concept of applying Bayesian decision theory in classification of multivariate data. Some of the previous authors are relying on nearest neighbor technique that is not reliable in large datasets [4, 8]. This paper, in contrast, eliminates any notion of "distance" entirely: statistical notions of object similarity were identified on the basis of class conditional and prior probabilities and were justified by using Bayesian Decision Theory with minimum error classification.

Overall, by demonstrating the utility of Bayesian Decision Theory, its comparison with Holland-style adaptive classifier, and implementation of standard dimensionality reduction method of PCA it can be concluded that both PCA and BDT can give efficient results in document analysis, which is no doubt the most rapid growing research area. This paper also allows us to aggregate different classification methods like multi-layer perceptrons or Linear Discriminat Analysis to improve the statistical pattern classification.

References

[1] P. W. Frey and D. J. Slate, *Letter Recognition using Holland-style adaptive classifiers*, Machine Learning, vol. 6, pp. 161-182 ,1991.

[2] Anil K. Jain, IEEE Member, Robert P.W. Duin and Jianchand Mao, Senior IEE Member. “Statistical Pattern Recognition: A Review”, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 22, January 2000

[3] Lindsay Smith. 2002. “A tutorial on Principal Component Analysis”. February 26, 2002.

[4] Thomas M. Breuel, “Character Recognition by Adaptive Statistical Similarity”, ICDAR 2003

[5] R. E. Schapire, Y. Freund, P. Bartlett and W. S. Lee, "Boosting the margin: A new explanation for *the* effectiveness of voting methods", Machine Learning: Proceedings of the fourteenth International Conference, pp. 322-330, 1997.

[6] H. Schwenk and Y. Bengio, "Adaptive Boosting of Neural Networks for Character Recognition", Technical report #1072, Department d' Informatique et Recherche Operationnelle, Universite de Montreal, Montreal, Qc H3C-3J7, Canada, 1997.

[7] R.O. Duda, P.E. Hart, “Pattern Classification” New York: John Wiley & Sons, 1998

[8] Kenji Fukumizu, Francis R. Bach, Michael I. Jordan. “Dimensionality Reduction for Supervised Learning with Reproducing Kernel Hilbert Spaces”, Journal of Machine Learning Research 5 (2004) 73-99 Submitted 5/03; Revised 10/03; Published 1/04

[9] Ethem Alpaydin. 2005. “Introduction to Machine Learning”. Prentice-Hall of India Private Limited, New Delhi under special arrangement of MIT Press USA. ISBN: 81-203-2791-8