INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

ICA an Efficient Algorithm for Feature Extraction

Divya Agarwal[1]

Department of Computer Science
Bhagwant Institute of Technology
Bijnore Uttar Pradesh, India
divya.agarwal1490@gmail.com

Abstract: *In this paper I have used an efficient algorithm for feature extraction i.e. Independent Component Analysis (ICA). By using ICA in solving supervised classification problems, we can get the new features which are independent from each other and which also convey the output information faithfully. The concept of ICA may actually be seen as an extension of principal component analysis (PCA) which can only impose independence up to the second order and consequently defines direction that are orthogonal. So by using ICA we can greatly reduce the dimensions of feature space without degrading the performance of classifying problems. Potential application of ICA includes data analysis and compression, Bayesians detection, localization of source, blind identification and deconvolution.*

Keywords: *extraction, perceptron, indented, component.*

1. INTRODUCTION

This paper considers the feature extraction problem since it often results in improved performance, especially when small dimensions are required, thus can be directly applied to make oblique decision borders as optimal classifiers like support vector machines do.[1] Recently, in neural networks and signal processing societies, independent component analysis (ICA), which was devised for blind source separation problems, has received a great deal of attention because of its potential applications in various areas. Though this ICA method can be directly applied to transform the original feature set into a new set of features as in [2]-[5], the performance cannot be guaranteed, since it does not utilize output class information like the PCA. These ICA-based feature extraction methods just focus on the visual representation of the data [6].

In this paper, I have used ICA-based feature extraction algorithm which utilizes the output class information in addition to having the advantages of the original ICA method. This method is well-suited for classification problems in the aspect of constructing new features that are strongly related to output class. By combining the proposed algorithm with existing feature selection methods, we can greatly reduce the dimension of feature space while improving classification performance.

2. FEATURE EXTRACTION USING ICA

The main idea of the proposed feature selection algorithm is very simple. In applying ICA to feature extraction, we include output class information in addition to input features. ICA is classified as unsupervised learning because it outputs a set of maximal independent component vectors. This unsupervised method by nature is related to the input distribution, but cannot guarantee good performance in classification problems though the resulting independent vectors may find some application in such areas as visualization [4] and source separation [3].Instead of using only the input vectors, we treat the output class information as one of the input features and append it to the inputs for ICA. To illustrate the advantage of this method, consider the following simple problem. Suppose we have two input features $x1$ and $x2$ uniformly distributed on [-1, 1] for a binary classification, and the output class y is determined as follows:

$$y = \begin{cases} 0 \ if \ x1 + x2 < 0 \\ 1 if \ X1 + x2 \geq 0 \end{cases}$$

Plotting this problem on a three dimensional space of ($x1$, $x2$, y) leads to Fig.1 where the class information as well as the input features correspond to each axis respectively. The data points are located in the shaded areas in this problem. As can be seen in the figure, this problem is linearly separable and we can easily pick out $x1 + x2$ as an important feature.

Performing ICA on a set of data provides us with *N*-dimensional vectors indicating the directions of independent components on feature space. If we take, The output class as an additional input feature, it forms an *(N+1)*-dimensional space, and applying ICA to this dataset gives us *(N+1)*-dimensional vectors.

These vectors are made to well describe the data distribution over the extended *(N+1)*-dimensional feature space and can be useful for classification problems if we project the vectors onto the original N-dimensional feature space. In the problem above, the vector shown as a thick arrow in Fig.1 will surely indicate the apparent discontinuity in the data distribution over the extended feature space. This vector can provide us with a great deal of information about the problem and projecting it onto the (x1, x2) feature space gives us a new feature relevant to the output class.
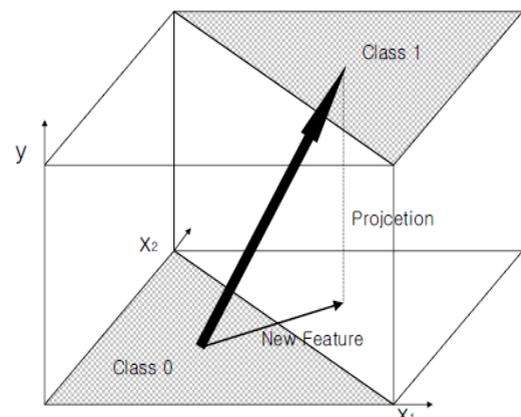

Fig.1: Concept of ICA based feature extraction

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

Based on this idea, a feature extraction algorithm for classification problem is used as follows:

2.1. Preparing data

(a) With N input features x= [x□, · · · , xN] T and one output class c, compose a new (N+1)-dimensional input dataset {(xT, c) T}.
(b) Normalize each feature fi by (fi−mi)/2σi where mi and σi are the mean and the standard deviation of fi, respectively.

2.2. Performing ICA

Apply ICA to the new dataset, and store the resulting weight matrix W of dimension (N+1) × (N+1).

2.3. Shrinking small weights

(a) For each N+1 independent row vector Wi of W, compute the absolute

$$\text{Mean}\,\alpha_i = \frac{1}{N+1}\sum_{j=1}^{N+1}\omega_{ij}$$

(b) For allwij W, if |wij | < α · ai, then shrink |wij | to zero. Here, α is a small positive number 1.

2.4. Extracting feature candidates

(a) For each weight vector Wi, project it onto the original input feature space, i.e., delete weights wic (= wi, N+1) corresponding to the output class, resulting in new N-dimensional row weight vector W_i.
(b) By multiplying new weight matrix W_ of dimension (N+1) × N to the original input data x, construct a (N+1)-dimensional vector whose components fi's are new feature candidates. 1
In the simulations, setting α = 0.2 or 0.5.

2.5. Deleting inappropriate features

(a) Form a set of feature candidates F = {fi = W_ ix, i∈ 1 · · · N+1}.Then set FS = F.
(b) For each feature candidate fi, if the corresponding weight for class which is zero, then exclude it from FS.
(c) For each feature candidate fi, if corresponding weights wij = 0 for all j ∈ 1 · · · N, then exclude fi from FS.
(d) Resulting FS contains final N_ extracted features.
Step 3 is useful for excluding the effect of features which do not contribute significantly to the new features. The effect of the exclusion is controlled by a suitable choice of α.
When we applied the algorithm to the problem illustrated in Fig.1, with 1,000 samples, we obtained 3 raw weight vectors W = {(w1, w2, wc)} = {(7.55, 1.74,−2.24), (2.10, 7.88,−2.48), (−0.75,−0.78, 3.53)}.Note that the last one leads to a new feature −0.75x1−0.78x2 which is nearly equivalent to the optimal
Feature x1 + x2. [7] Though we consider only two-class problems in the algorithm, the algorithm can easily be extended to multi-class problems which can be set up as a series of two-class problems, where in each two-class problem, one class is distinguished from the others.

3. EXPERIMENTAL RESULTS

In this section I have applied the proposed algorithm on UCI dataset [8] and will present some experimental results which show the characteristics of the proposed algorithm for the classification problems in this section, to show the effectiveness of the proposed algorithm, we selected appropriate numbers of original features and extracted features by MIFS-U [9] and compared the classification performance. We tested the proposed algorithm for several problems, but present the results of only two cases for space limitation. The other cases show the similar enhancement in performance. Wisconsin Breast Cancer Diagnosis. The dataset consists of nine numerical attributes and two classes which are benign and malignant. It contains 699 instances with 458 benign and 241 malignant. There are 16 missing values and in my experiment I have replaced these with average values of corresponding attributes. The proposed algorithm extracted seven new features. I select appropriate numbers of original features and extracted features by MIFS-U [9] and trained the data with C4.5 [10] and multilayer perceptron with standard back propagation (BP).MLP was trained for 500 iterations with three hidden nodes, zero momentum and learning rate of 0.2. For verification, 10-fold cross validation is used. The classification results are shown in Table 1 and the numbers in parentheses are the tree sizes of C4.5.

Table 1
Classification performance for Breast Cancer data (%)

No. of features Selected	Original Features (C4.5/MLP)	Extracted Features C4.5/MLP
1	91.13(3)/92.42	95.42(3)/95.42
2	94.71(13)/96.28	95.57(9)/95.42
3	95.85(23)/96.14	95.71(11)/96.57
9	94.56(31)/96.42	–

Table 2: Classification performance for Chess data (%)

No. of features Selected	Original Features (C4.5/MLP)	Extracted Features C4.5/MLP
1	67.43	96.23
2	74.52	97.98
3	91.08	98.80
36	98.54	–

[11] It shows that with only one extracted feature, we can get near the maximum classification performance that can be achieved with at least two or three original features. Also, note that the tree size does not expand much for the case of extracted features, which is desirable for generating simple rule. Chess End-Game Data. There are 36 attributes with two classes, white-canwin ('won')/white-cannot-win ('nowin'). There are 3,196 instances of which 1,669 are 'won' and the

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

other 1,527 are 'nowin' with no missing values. In conducting feature extraction algorithm, we set $\alpha = 0.5$ and obtained six additional features. After feature selection process with MIFS-U, we trained the data for 500 iterations by the standard BP with 0.2 learning rate and zero momentum. I divided the data into two sets, one half for training and the other half for testing. Table 2 shows the classification performances on the test data. The used method shows significant improvement in performance over the original features.

4. CONCLUSION

ICA has the ability to generate appropriate features for classification problems. As the name indicates, ICA is characterized by its ability of identifying statistically independent components based on input distribution. Although ICA can be directly used for feature extraction, it does not generate useful information because of its unsupervised learning nature. In this algorithm, The added class information plays a critical role in the extraction of useful features for classification. F or problems Feature Extraction Using ICA 573 with original N features, we get N+1 weight vectors in (N+1)-dimensional extended feature space. Projecting them onto original feature space leads to new feature candidates and by deleting inappropriate ones which have little correlation.

With the class information, we can extract better features. Since it uses the standard feed-forward structure and learning algorithm of ICA, it is easy to implement and train. If we use the proposed algorithm with the conventional feature selection algorithm, we can get simpler classifier with better performance. Experimental results for several datasets show that the ICA algorithm generates good features which outperform the original features for classification problems. Though the algorithm is proposed for two-class problems, it is also useful for multi-class problems with simple modification.

REFERENCES

[1] V. Cherkassky and F. Mulier, Learning from Data: Concepts, Theory, and Methods, pp. 60-65, John Wiley & Sons, 1998.

[2] Dorffner, H. Bischof, and K. Hornik (Eds.): "Feature Extraction Using ICA"ICANN 2001, LNCS 2130, pp. 568–573, 2001.

[3] H. Liu and H. Motoda, "Less is more," Feature Extraction Construction and Selection,pp. 3-11, Boston: Kluwer Academic Publishers, 1998.

[4] A.J. Bell and T.J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," Neural Computation, vol 7, pp. 1129-1159, 1995.

[5] A. Hyvarinen, E. Oja, P. Hoyer and J. Hurri, "Image feature extraction by sparsecoding and independent component analysis," Fourteenth International Conferenceon Pattern Recognition, vol. 2, pp. 1268 -1273, 1998.

[6] A.D. Back and T.P. Trappenberg, "Input variable selection using independent component analysis," The 1999 Int'l Joint Conf. on Neural Networks, July 1999.

[7] H.H. Yang and J. Moody, "Data visualization and feature selection: new algorithms for nongaussian data," Advances in Neural Information Processing Systems, vol 12, 2000.

[8] T.M. Cover, and J.A. Thomas, Elements of Information Theory, John Wiley &Sons, 1991.

[9] P.M. Murphy and D.W. Aha, UCI repository of machine learning databases, 1994. For information contactmlrepository@ics.uci.eduorhttp://www.cs.toronto.edu/~delve/

[10] N. Kwak and C.-H. Choi, "Improved mutual information feature selector for neural networks in supervised learning," The 1999 Int'l Joint Conf. on Neural Networks, July 1999.

[11] R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA.,1993