SCENIC BEAUTY ESTIMATION USING INDEPENDENT COMPONENT ANALYSIS AND SUPPORT VECTOR MACHINES

X. Zhang, V. Ramani, Z. Long, Y. Zeng, A. Ganapathiraju and J. Picone
Institute for Signal and Information Processing
Mississippi State University
Mississippi State, Mississippi 39762, USA
e-mail: {zhang, ramani, long, zeng, ganapath, picone}@isip.msstate.edu
Ph (601) 325-3149 - Fax (601) 325-3149

Abstract **- The objective in the Scenic Beauty Estimation (SBE) problem is to develop an automatic classification algorithm that matches human subjective ratings. Algorithms such as Principal Components Analysis (PCA) and Decision Trees (DT) have been applied to this problem with limited success, motivating our search for a better classifier. Since this is obviously a nonlinear classification problem, we applied two nonlinear techniques: Independent Component Analysis (ICA) and Support Vector Machines (SVMs). We evaluated these algorithms on a standard, publicly available data set using a variety of combinations of features. The optimally configured ICA and SVM systems achieved misclassification rates of 33.4% and 32.2% respectively. This is a significant improvement over the best results previously reported on this task: 36.6% for PCA and 43% for DT. Since ambiguity in the features space is a significant problem in this application, these results validate the effectiveness of nonlinear classification techniques.**

INTRODUCTION

The United States Forest Service (USFS) has a long-term interest in the development of automatic methods [1] for managing forest resources. These methods use a database of forestry images to determine the utility of a plot of forest land both in terms of timber use and scenic quality. Traditional methods used to determine the scenic quality are very tedious and involve a large group of people manually rating each of the images. Clearly, an automatic method has advantages that include consistency and efficiency.

To facilitate this research, an extensive database of images [2] has been developed, along with a comprehensive evaluation paradigm. A wide array of features were extracted from these images, and several techniques, ranging from Principal Components Analysis (PCA) to Decision Trees (DT) were used to combine these features. Unfortunately, despite the incorporation of such powerful statistical measures, overall performance [3] on this task was not much better than chance.

ALGORITHM DESCRIPTION

Linear classifiers [4] are often used due to the simplicity of implementation, and their robustness to poorly estimated statistical models. A linear classifier uses a discriminant function that can be represented as:

$$(h(X) = V^T X + v_o) \gtrless 0 \quad , \qquad (1)$$

which essentially describes a hyperplane decision region separating two classes. However, for many applications, the classes cannot be separated by a hyperplane, and a nonlinear classifier is required to achieve good performance.

Principal Components Analysis

The most straightforward method of classification using a linear classifier is to use a statistical normalization approach such as PCA [4]. Here, it is assumed that the first principal component of a sample vector lies parallel to the direction along which there is the largest variance over all samples. This direction corresponds to the eigenvector associated with the largest eigenvalue. The k^{th} principal component is chosen to be the linear combination of the input features that has the largest variance, under the constraint that it is also uncorrelated to the previous *k-1* principal components. This approach, depicted in Figure 1, works well when the direction along which there is maximum variation also contains the information about the class discrimination.

Independent Components Analysis

The principle behind ICA [5] is very similar to PCA with the main difference being that the resulting transformation no longer be orthogonal. This is depicted in Figure 1. Similar to PCA, ICA finds a linear coordinate system for the data, transforming it using a linear matrix transformation. Hence, the complexity of the run-time analysis is equivalent to PCA.

However, unlike PCA, the transformation is free to be non-orthogonal, as shown in Figure 1. Furthermore, the

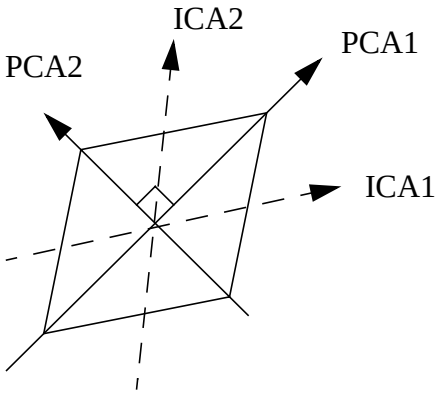

Figure 1. A comparison between transformations found by PCA and ICA when the data is uniformly distributed within the diamond-shaped region.

computation of this transform involves higher order statistics of the data. In the PCA case, which involves an underlying assumption that the data is multivariate Gaussian, only the mean and covariance (second-order statistics) are used.

Our approach to finding the ICA transformation follows that described in [6]. To find a linear transformation, $\overline{W}$, we minimize the mutual information of the transformed data. Since a closed-form solution is not tractable, we use a stochastic gradient ascent algorithm. When the nonlinear function is the same (with respect to scaling and shifting) as the cumulative density functions of the underlying independent components, it can be shown that a nonlinear infomax procedure also minimizes the mutual information between the components of the transformed data.

However, in practice, we must pick a nonlinear function for the stochastic ascent without any detailed knowledge of the probability density functions of the underlying independent components. Our procedure is to maximize $H[g(u)]$, where $g(\mu)$ is a sigmoidal function, and to assume the pdf's of μ are super-Gaussian (similar to a Gaussian, but more peaky with longer tails). This approach leads to an ICA solution when such a solution exists.

The specific form of the stochastic gradient ascent that is used involves a learning rule that changes weights according to the gradient of the entropy [6]:

$$\Delta W \propto \frac{\partial H(y)}{\partial W} W^T W = (I + \hat{y} u^T) W \qquad (2)$$

in which $\hat{y}_i = \frac{\partial}{\partial y_i}\frac{\partial y_i}{\partial u_i} = \frac{\partial}{\partial u_i} In \frac{\partial y_i}{\partial u_i}$ is computed as $\hat{y}_i = 1 - 2y_i$, and $y_i = (1 + e^{-u_i})^{-1}$.

Support Vector Machines

A support vector machine (SVM) [7] is a new technique for classification. The main objective of this technique is to construct an optimal hyperplane, which uses a small part of training set as support vectors. If the training vectors are separated without errors by an optimal hyperplane, the expected error rate on a test sample is bounded by the ratio of the expected number of the support vectors to the number of training vectors. Since this ratio is independent of the dimension of the problem, good generalization is guaranteed if we can find a small set of support vectors.

First, we will have a look at the simplest case: linear machines trained on separable data. Suppose we have 2 classes. Label the training data as $\{x_i, y_i\}$, $i = 1, \ldots, l$, $y_i \in \{-1, 1\}$, $x_i \in R^d$, we would like to find a separating hyperplane $w \cdot x + b = 0$, where w is normal to the hyperplane. This hyperplane separates the positive examples from the negative ones. Let d_+ (d_-) be the shortest distance from the separating hyperplane to the closest positive (negative) example, then $d_+ + d_-$ is called the "margin" of the separating hyperplane.

Here, the support vector machine just looks for the separating hyperplane with the largest margin. This has been shown [8] to be equivalent to the following problem. Given the inequalities:

$$y_i(x_i \cdot w + b) - 1 \geq 0 \qquad \forall i \qquad (3)$$

find hyperplanes of H_1: $x_i \cdot w + b = 1$ and H_2: $x_i \cdot w + b = -1$, which give the maximum margin by minimizing $||w||^2$, subject to the constraints in Eq. 3.

To solve the problem using Lagrangian method, we have to maximize $L_D = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j$ with respect to α_i, subject to constraints $\sum_i \alpha_i y_i = 0$ and the positivity of α_i [8]. The solution is given by $w = \sum_i \alpha_i y_i x_i$.

Once a Support Vector Machine has been trained in this way, a test vector x may be classified by simply determining on which side of the decision boundary hyperplane it falls.

For non-separable data, the constraints in Eq. 1 should be modified as follows:

$$x_i \cdot w + b \geq 1 - \xi_i \qquad y_i = 1 \tag{4}$$

$$x_i \cdot w + b \leq -1 + \xi_i \qquad y_i = -1 \tag{5}$$

$$\xi_i \geq 0 \qquad \forall i \ . \tag{6}$$

The solution is similar to that described previously.

The above SVM algorithm on linear data set can be extended to nonlinear data. The general idea is that we can map the nonlinear data into another space in which we can still do a linear separation on the mapped data. To do the mapping, we need to find a suitable "kernel function." A detailed discussion of this approach is found in [7].

APPLICATION TO SBE ESTIMATION

A detailed overview of the SBE Estimation problem can be found in [2]. We have converted the SBE problem to a three-way classification problem. Our implementation of ICA was based on information maximization theory. If the output entropy is maximized, then the components of the output vector must be statistically independent.

Assuming we have a training set $\{x\}$. We need to find a transformation $u = \overline{W}x$, with the properties mentioned above. First, the data x is prewhitened by $\{x\} \leftarrow 2W_z(\{x\} - \langle x \rangle)$, where $W_z = \langle xx^T \rangle^{-1/2}$. The matrix W is then initialized to the identity matrix, and trained accordingly.

We use a context-dependent transformation in this case, Hence, three separate transformations are trained, and the test images are classified by using a class-specific Euclidean distance computation in the transformed space.

Our SVM implementation follows a similar strategy. One classifier is trained for each of the three classes. Each classifier is trained by considering the in-class data versus the rest of the data. For each test image, we calculate the distance of its feature vector to each of the three classifiers. Classification was achieved by comparing the distances of each image's vector from the three hyperplanes.

The SVM implementation was accomplished using the public-domain *SVMlight* [8] package. The radial basis function kernel was chosen because it was found to give best performance on a number of related classification tasks.

PERFORMANCE

Both ICA and SVMs were evaluated on a standard subset of 637 images. There are three classes of images in this subset: low, medium, and high scenic beauty. A total of 45 features are extracted from the images including color content, density of long lines, entropy and fractal dimension. We examined the following combinations of features:

- ALL: All 45 features
- RGB: Red + Green + Blue
- RGB+LL: RGB + Long Lines (postprocessed from a standard edge detection algorithm)
- RGB+LL+ENT: RGB+LL + Entropy
- RGB+LL+FRACT: RGB+LL+ENT + Fractal Dims.

The results are shown in Tables 1 and 2 below. The 637 image dataset is partitioned four different ways to provide

features	Error Rate (%)					
	set1	set2	set3	set4	mean	vari-ance
ALL	35.2	33.5	31.9	33.1	33.4	5.8
RGB	34.6	33.5	32.5	33.1	33.4	2.3
RGB+ LL	34.6	33.5	33.1	33.1	33.6	1.4
RGB+ LL+ENT	34.6	34.2	33.1	32.5	33.6	2.8
RGB+ LL+FRA	35.9	33.5	32.5	33.1	33.8	6.4

Table 1. Performance of ICA on four evaluation sets.

features	Error Rate (%)					
	set1	set2	set3	set4	mean	vari-ance
ALL	34.6	34.2	31.2	31.9	33.0	8.4
RGB	35.2	34.8	32.5	32.5	33.8	6.3
RGB+ LL	32.1	32.3	31.9	32.5	32.2	0.2
RGB+ LL+ENT	35.2	34.2	31.9	32.5	33.4	6.9
RGB+ LL+FRA	35.2	35.2	32.5	31.2	33.5	12.1

Table 2. Performance of SVM on the same four sets.

four separate evaluations.

A comparison of ICA and SVM to existing state-of-the-art is shown in Figure 2. ICA and SVM provide improved performance over PCA. Neither ICA nor SVM is better on all experimental conditions, however. This suggests there is still some work ahead on dimensionality reduction and feature set enhancement.

CONCLUSION

We have applied two new nonlinear classification techniques, ICA and SVM, to the SBE problem. We have shown that both ICA and SVM perform better than standard classification schemes such as PCA. Overall performance, however, is still far from human performance.

The best error rate achieved by ICA is 33.4%, using a simple combination of R, G, and B. This is extremely promising in that it utilizes the simplest feature vector, and seems to learn all other derived information, such as entropy and fractal dimension. SVMs, on the other hand, seem to make effective use of long line information in combination with RGB.

These results are superior to the best results we have obtained with alternate techniques: 36.6% for PCA, and 43% for Decision Trees. Encouraged by these results, we believe that we can combine these two techniques to produce a better classification scheme. By using ICA as a front-end to SVMs, we will be able to supply optimally separated distributions to the SVMs, thereby constructing a better nonlinear decision region. We expect that the classification capability of the combined system will be a great improvement over the system using either technique independently. It is important to note that, as encouraging as these numbers are, the performance of intelligent guessing (always guess msbe) is still extremely close to our best classification system.

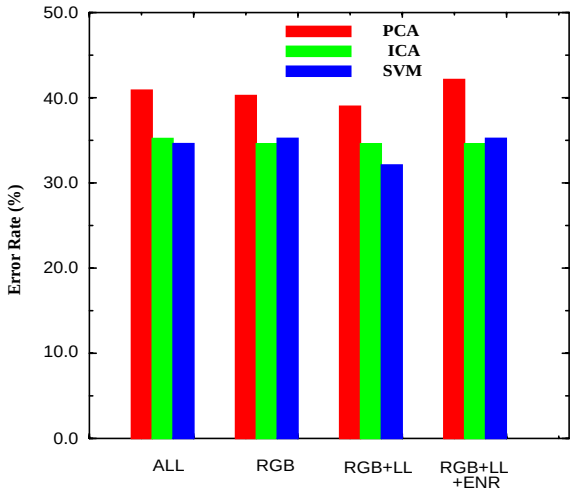

Figure 2. A comparison between PCA, ICA and SVM on evaluation set no. 1 of the USFS SBE database.

REFERENCES

[1] N. Kalidindi, A. Le, L. Zheng, H. Yaquin, V. Rudis and J. Picone, "Scenic Beauty Estimate of Forestry Images," *Proceedings of the IEEE Southeastcon*, pp. 337-339, Blacksburg, Virginia, USA, April 1997.

[2] N. Kalidindi, "Scenic Beauty Estimation of Forestry Images," *http://www.isip.msstate.edu/publications/seminars/masters_oral/1997/usfs_ imaging/index.html*, Master of Science Special Project Presentation, Mississippi State University, MS State, MS, USA, November 1997.

[3] J. Ngan, "Information Theory Based Decision Trees for Data Classification," *http://www.isip.msstate.edu/publications/seminars/masters_oral/1998/decision_tree_c4/index.html*, Master of Science Special Project Presentation, Mississippi State University, MS State, MS, USA, December 1997.

[4] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, San Diego, California, USA, 1990.

[5] T. Bell, "Source Separation and Learning Non-orthogonal Bases for Signals Using Independent Component Analysis," *http://www.clsp.jhu.edu/ws98/presentations/workshop/bell/abstract.html*, presented at the 1998 Summer Workshop on Language Engineering, Center for Language and Speech Processing, John Hopkins University, Baltimore, Maryland, USA, July 1998.

[6] A. J. Bell and T. J. Sejnowski, "An Information-Maximization Approach to Blind Separation and Blind Deconvolution," *Technical Report No. INC-9501*, Institute for Neural Computing, University of California at San Diego, San Diego, California, USA, February 1995.

[7] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer Verlag, New York, New York, USA, 1995.

[8] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *http://svm.research.bell-labs.com/SVMdoc.html*, Bell Laboratories, Lucent Technologies, Holmdel, New Jersey, USA, December 1998.