Face Recognition Using ICA and Class Information

Lidan Zhang, Fenggang Huang, Xianwei Li

Department of Computer Science, Harbin Engineering University, Harbin, Heilongjiang, China

Abstract In this paper we modify the basic ICA algorithm by utilizing class information, and then apply it to the field of face recognition. Firstly, we address the face representation used in ICA and adopt an architecture whose outputs are independent or as independent as possible. Secondly, we present our modified ICA algorithm. By minimizing the distance within class we derive a simpler and faster algorithm, which is more suit for recognition. Three public available databases (UMIST, ORL and Yale University) are selected to evaluate the recognition performance and computational cost. Experiments show that the results are encouraging.

Keywords: Independent Component Analysis; Face recognition; Class information

1. Introduction

Face recognition by machine has been started since 70s and becomes an active and important research area. Since then, varieties of new methods have been developed, among which subspace analysis techniques have absorbed a growing interest. Commonly, most of the subspace techniques deal with a problem that is dimension reduction. It means that we project the images into the lower space (so-called face space) due to the consideration of computational feasibility and efficiency. Therefore, dimension reduction techniques such as Principal Component Analysis (PCA)[1], and Independent Component Analysis (ICA) are presented and become two important methods for face global representation.

PCA is the basic method of modeling the high-dimension data. It projects the original images into a lower dimension space, which is spanned by the bigger eigenvectors of the data's covariance matrix. But PCA method starts from the purpose of image reconstruction not recognition, because it minimizes the square error between he original faces and the coded ones.

ICA is a generalization of PCA that uses the high-order moments of the input images other than the second-order moments. It transforms an observed multi-dimensional vector into components that are statistically as independent from each other as possible. Theoretically, ICA is advantaged than PCA. But some experiments showed that there is no performance difference between ICA and PCA[2.3].

As mentioned above, in some cases ICA is disadvantage to PCA. We think the reason may be that the original ICA algorithm only uses the data's statistical information and neglects the sample's class information. The goal of our algorithm is to modify the original ICA by making the use of the class information. That is, we minimize the distance within class when we try to find the independent components. Experiments have showed that the attempt is successful.

The rest of this paper is organized as follow. Section 2 introduces the original ICA algorithm briefly and our architecture to represent face of ICA. In Section 3 we give our modified ICA algorithm, also we give some experimental results on Section 4. At last, Section 5 concludes the paper.

2. A brief introduction to basic ICA algorithm

ICA was originally developed for separating mixed audio signals into independent sources[4]. The ICA model is:

$$X=AS \qquad U=WX \tag{1}$$

The above two equations individually represent mixing and demixing process. Here, X and S are respectively observed variables and hidden sources; A and W are respectively unknown mixing matrix and demixing matrix.

There are many different methods to evaluate the independence and some corresponding algorithms. The famous one is the fixed-point algorithm[5] based on the maximization of the negentropy. Hyvärinen referred that negentropy is an efficient measure of independence; also he introduced a flexible and reliable approximation of the negentropy, that is:

$$J_G(x) \approx \rho[E\{G(x)\} - E\{G(v)\}]^2 \qquad (2)$$

where ρ is a positive constant, we often set ρ=1. v is a Gaussian variable having zero mean and unit variance. Correspondingly, we can solve the ICA problem by the following optimization problem:

$$\begin{aligned} &maximize \sum_{i=1}^{n} J_G(w_i) \qquad w_i = 1,\ldots,n \\ &subject\ to \quad E\{(w^T x)^2\} = 1 \end{aligned} \qquad (3)$$

We can see that ICA doesn't use the data's class information and may not be good for classification. From this point, we try to modify ICA of this paper and take the data's label information into the consideration.

2.1. Face Representation of ICA

The architecture[6] used in this paper(called factorial code) is to make the coefficients, which linearly combined to build test face, statistically independent. So the face vectors are dealt as the columns of X. The rows of W were a set of basis images for the faces, and the columns of the ICA output U are factorial codes of the faces.

Denote u_i as the code of the i-th face. So we can measure the scatter degree within kth-class after ICA:

$$\begin{aligned} D &= W^T S_k W \\ S_k &= \sum_{x_i x_j \in k} (x_i - mean_k)(x_j - mean_k)^T \end{aligned} \qquad (4)$$

S_k is the within class scatter matrix of the class k which has been used in fisher criterion, $mean_k$ is the mean value of class k. From the point of classification, we hope D to be minimized. This can be used later as a constrained condition for ICA, in order to make ICA be suit for classification.

3. ICA using class information

Here we bring our modified ICA algorithm based on one unit. Firstly consider the constrained condition, which has discussed above. For One-Unit ICA, this condition can be modified as $D_{1\text{-unit}}=w^T S_k w$($w=w_i$, i=1,…,n for simplicity).We want to minimize the $D_{1\text{-unit}}$, so we can set a threshold ε to make:

$$d(w) = w^T S_k w - \varepsilon \le 0 \qquad (5)$$

when and only when w is the optimal vector, that $d(w)=0$. Until now, we can modify the ICA's optimal problem as:

$$\begin{aligned} &maximize\ J_G(w) = [E\{G(w)\} - E\{G(v)\}]^2 \\ &subject\ to\ d(w) = w^T S_k w - \varepsilon \le 0\ , \\ &\qquad h(w) = E\{(w^T x)^2\} = 1 \end{aligned} \qquad (6)$$

Notice that the maximum of $J_G(w)$ are obtained at certain optima of $J(w)=E\{G(w^T x)\}$. Also, introduction a slack variable ξ , the augmented Lagrangian function $L(w,\mu,\lambda)$ is given by:

$$\begin{aligned} L(w,\mu,\lambda) = &E\{G(w^T x)\} - \frac{1}{2r_p}[\max^2\{\mu + r_p d(w), 0\} \\ &- \mu^2] - \lambda h(w) - \frac{r_p}{2}\|h(w)\|^2 \end{aligned} \qquad (7)$$

where μ, λ are Lagrangian multipliers for constraints $d(w)$ and $h(w)$, respectively, r_p is scalar penalty parameter, the last item of equation(7) is the penalty item to ensure that the optimization problem is held at the condition of local convexity assumption. To find the maximum of L , w can be learned through a Newton-like learning method:

$$w^+ = w - \eta L'/L'' \qquad (8)$$

where η is a positive learning rate added to avoid the uncertainty in convergence. Consider the data is whitened ($E\{x^T x\}=I$), we can obtain that:

$$L' = \frac{\partial}{\partial w} L = E\{xg(w^T x)\} - 2(\mu + r_p w^T S_k w) S_k - \lambda w$$
$$L'' = \frac{\partial}{\partial w} L' = E\{g'(w^T x)\} - 2[(\mu + r_p w^T S_k w) S_k + 2 r_p S_k w (S_k w)^T] - \lambda \quad (9)$$

Here, S_k is the within class matrix of the class which w belonged. Where μ, λ are updated according:

$$\mu^+ = \max\{\mu + r_p g(w), 0\}$$
$$\lambda^+ = \lambda + r_p h(w) \quad (10)$$

Now, we have got the learning algorithm. By simplify, let η=1 and use a more heuristic derivation[7], we get the simpler form of Modified ICA(M-ICA):

$$w^+ = E\{xg(w^T x)\} - E\{g'(w^T x)\} w + r_p (S_k w)(S_k w)^T w$$
$$w^* = w^+ / \left\| w^+ \right\| \quad (11)$$

Here, r_p can be considered as a control of the interference quality of the class information..

4. Experiments

Our experiments are carried out on three faces database: ORL, UMIST and Yale face database, to test our algorithm's recognition performance and computational time. Next, let's briefly introduce the three standard face databases.

4.1. Database

- **ORL face database:** The database consists of 400 images acquired from 40 persons with variations in facial expression and facial details. All the subjects were in an upright frontal position, with tilting and rotation tolerance up to 20 degree, and tolerance of up to about 10% scaly. In our experiments, five images are randomly chosen from the ten images available for each subject for training, while the remaining five images are used for testing.
- **Yale face database:** The database contains 165 images with 11 different images for each of the 15 distinct subjects. We randomly take 5 images of each subject for training and leave the rest for testing.

Table 1. Face recognition performance using PCA, FastICA and M-ICA based on three different distance measures: L1(L_1 distance measure), L2(the L_2 distance measure) and Cos(the cosine similarity measure)

%		UMIST	ORL	Yale	Average
PCA	L1	86.40	86.00	85.56	86.17
	L2	86.40	91.00	86.67	87.82
	Cos	86.67	90.50	83.33	87.37
FastICA	L1	83.73	65.00	75.56	77.00
	L2	85.07	64.50	75.56	77.59
	Cos	85.87	84.50	88.89	85.86
M-ICA	L1	86.67	92.50	85.56	88.42
	L2	86.67	92.50	85.56	88.27
	Cos	87.20	91.00	88.89	88.57

- **UMIST face database:** The database consists of 575 gray-scaled images of 20 subjects, each covering a wide range of poses from profile to frontal views. Also, we take a half of images and left the remainders as test samples.

4.2. Comparison of recognition accuracy

The PCA algorithm we used to contrast is the standard one and the ICA algorithm is FastICA. Also, the simpler nearest neighbor classifiers based on different distances are used to valuate the recognition accuracy, because the focus of the paper is on feature extraction. The distances we use here are L1 norm, L2 norm and cosine angle. The face recognition accuracies of PCA, ICA and M-ICA on the three databases that are described above are given in Table 1. Average face recognition accuracy of PCA, FastICA and M-ICA. Note here that the average recognition rate is calculated on the total mistaken and right classified sample numbers of the three databases.

From the table, we can see that M-ICA algorithm is better than PCA and ICA. It is due to the inference of class information. From the experiments of M-ICA, we anticipate that if we use more class information, such as distance between classes, the classification

accuracy will be improved. Also we can see from table1 that, other than ICA, M-ICA is robust to the selection of the distance measurement.

4.3. Computation cost

Computation cost is another important issue for face recognition, especially for practical problem. So, we compare the training time of ICA and M-ICA.

According to the algorithm, we can see that the most of the time is spent on the iteration to find optimal w. Also from equal (11), the computation of an iteration of M-ICA is almost the same as the ICA. So for testing the computation cost here, we compare the iteration numbers for finding an IC. In addition, because the FastICA and M-ICA are both approximate approaches, the iterative numbers are not same even for same training samples. So we run 5 times on every database and get the average iterative number. The total average is carried on the three databases. This is shown on table2:

Table 2. the number of iterations for finding a IC

	UMIST	ORL	Yale	Average
FastICA	39.98	39.96	41.15	40.15
M-ICA	23.76	22.47	17.20	23.45

From the above table, we can see that the iterative numbers per IC of M-ICA is much smaller than FastICA. Furthermore, the training time of M-ICA is much smaller than FastICA. This is the result of preprocessing of M-ICA.

5. Conclusion

In this paper we use facial code to represent face and modify the ICA algorithm. Three standard public face databases have been used to evaluate the performance of our proposed method. Experimental results have shown that the recognition rate is improved and the computation load for calculating ICs is greatly reduced.

Further studies will investigate incorporating more class information into ICA architecture, such as the distance between classes and develop it into a batch version. In addition, we can see that our algorithm is only an approximation. We think the accuracy can be even higher if we can get a more accurate computation.

References

[1] M. Turk and Pentland, Face Recognition Using Eigenfaces, in Proc. IEEE International Conference on Computer Vision and Pattern Recognition, Maui, Hawaii, 1991

[2] Moghaddam, B. Principal Manifolds and Bayesian Subspaces for Visual Recognition. In International Conference on Computer Vision. 1999, Corfu, Greece. P11316-1136

[3] Kyungim Baek, Bruce A. Draper, J. Ross Beveridge, Kai She, PCA vs. ICA: A comparison on the FERET data set

[4] Comon, P. Independent Component Analysis: A new concept? Signal Processing, 1994.36(3): p.287-314

[5] A. Hyvärinen and E. Oja, A fast fixed-point algorithm for independent component analysis. Neural Computation, 9: 1483-1492,1997

[6] Marian Stewart Barlett, H. Martin Lades, and Terrence J. Sejnowski, Independent component representations for face recognition, Conference on Human Vision and Electronic Imaging Ⅲ, San Jose, California, January, 1998

[7] A Hyvärinen. A family of fixed-point algorithms for independent component analysis. In Proc. IEEE Int. Conf. On Acoustics, Speech and Signal Processing (ICASSP'97), pages 3917-3920, Munich, Germany, 1997.