TWO-STAGE OPTIMAL COMPONENT ANALYSIS

Yiming Wu[†], *Xiuwen Liu*[†], *Washington Mio*[‡], *K. A. Gallivan*[*]

[†]Department of Computer Science,
[‡]Department of Mathematics,
[*]School of Computational Science,
Florida State University, Tallahassee, 32306, FL

ABSTRACT

Linear representations are widely used to reduce dimension in applications involving high dimensional data. While specialized procedures exist for certain optimality criteria, such as principle component analysis (PCA) and Fisher discriminant analysis (FDA), they can not be generalized for more general criteria. To overcome this fundamental limitation, optimal component analysis (OCA) uses a stochastic gradient optimization procedure intrinsic to the manifold giving by the constraints of applications and therefore gives a procedure for finding optimal representations for general criteria. However, due to its generality nature, OCA often requires extensive computation for gradient estimation and updating. To significantly reduce the required computation, in this paper, we propose a two-stage method by first reducing the dimension of input to a smaller one (but larger than the final resulting dimension) using a computationally efficient method and then performing OCA in the reduced space. This reduces the computation time from days to minutes on widely used databases, making OCA learning feasible for many applications. Additionally, since the reduced space is much smaller, the stochastic gradient optimization tends to be more efficient. We illustrate the effectiveness of the proposed method on face classification.

Index Terms- Machine Vision, Face Recognition, Image Analysis, Optimal Method, Stochastic Process

1. INTRODUCTION

Recognition of objects using statistical methods from the 2D images is commonly adopted. However, as images are, in general, rather high dimension, the recognition tasks become computationally expensive and often infeasible as statistical methods are often limited to low dimensional data. On the other hand, it is well-known that images are generated by imaging processes with typically a small number of physical parameters such as lighting, orientation, etc. Thus, one way to overcome this problem is to reduce the input dimension while, at the same time, preserving most image information. A commonly used method is to project images linearly into a low dimensional subspace and use this projection for processing.

In recent years, several methods have been used on such dimension reduction problems, including Principle Component Analysis (PCA)[1] Fisher Discriminative Analysis (FDA) [2] and Independent Component Analysis (ICA) [3], etc. PCA is a linear unsupervised method which retains the maximum amount of variance within the projected feature space. However, when applied to the classification problem, the main drawback of PCA lies in the fact that class information is not utilized for class projection because PCA choose axes only based on the variance of data. ICA is a more general method than PCA which finds the independent components by maximizing the statistical independence of the estimated components. The classical FDA aims to find an optimal basis by minimizing the within-class distances and maximizing the between-class distance simultaneously, thus achieving maximum discrimination. An intrinsic limitation of FDA is that the formulation is based on the assumption that the underlying probability distribution for each object is Gaussian with the same variance; however, it has been shown that distributions of images are highly non-gaussian and thus the optimality of FDA in general for recognition is not guaranteed. Another limitation of classical FDA is that its objective function requires that one of the scatter matrices be nonsingular. However, in many real applications, such as face recognition and text classification, the scatter matrix in question can be singular since the dimension of data, in general, exceeds the number of data points. This is known as *singularity* problem. Several methods which extends FDA have been proposed recently to deal with this problem. PCA+LDA [4] applies PCA on the original data to obtain a more compact representation so that the scatter matrix becomes nonsingular. LDA/QR method [5] solves this problem by first applying QR decomposition to a small matrix involving the class centroid, and then LDA method is used in the reduced space.

Unlike these methods, Optimal Component Analysis (OCA) [6] [7] is a recently proposed stochastic method which can be applied to dimensional reduction and pattern recognition. The search for optimal linear representation is based on a stochas-

tic optimization process which maximizes a pre-specified performance function over all subspaces of a particular dimension. Its effectiveness has been demonstrated on a number of applications. However, a major limitation of OCA that prevents its wide use is its computation cost. In this paper, we propose a two-stage OCA (2-OCA) method to overcome this limitation. Our goal is to reduce the computation cost while at the same time to maximize the performance. The first stage of our proposed method is to obtain a more compact representation of the input images by dimensional reduction, then, in the second stage, OCA searching is conducted in this reduced space. As the search space is much lower compared with the original search space, the searching time is reduced dramatically and the performance in typical applications is kept at the same time.

The rest of the paper is organized as follows: Section 2 gives a review of OCA method; then the proposed two-stage OCA method is presented at Section 3; A comparative study of the performance of the 2-OCA method is given in Section 4; Section 5 concludes the paper with a brief summary.

2. REVIEW OF OCA

For recognition applications based on 2-norm distances, OCA provides a general subspace formulation on Grassmann manifold and a stochastic optimization algorithm is applied to computing the optimal basis. Comparing to PCA, ICA and FDA, OCA has been shown to have advantages in solving object recognition problems on some datasets. More specifically, in [6], the performance function F is defined in the following way. Let there be C classes to be recognized from the images; each class has k_{train} training images (denoted by $I_{c,1}, \ldots, I_{c,k_{train}}$) and k_{test} test images (denoted by $I'_{c,1}, \ldots, I'_{c,k_{test}}$) to evaluate the recognition performance measure.

$$F(U) = \frac{1}{Ck_{test}} \sum_{c=1}^{C} \sum_{i=1}^{k_{test}} h(\rho(I'_{c,i}, U) - 1). \tag{1}$$

where h(·)is a monotonically increasing and bounded function. In our implementation, $h(x) = 1/(1 + \exp(-2\beta x))$ where β controls the degree of smoothness of $F(U)$ and

$$\rho(I'_{c,i}, U) = \frac{\min_{c' \neq c, j} d(I'_{c,i}, I_{c',j}; U)}{\min_j d(I'_{c,i}, I_{c,j}; U) + \epsilon}. \tag{2}$$

Here

$$d(I_1, I_2; U) = \|\alpha(I_1, U) - \alpha(I_2, U)\|, \tag{3}$$

and $\|\cdot\|$ denotes the 2-norm, $\alpha(I, U) = U^T I$, and $\epsilon > 0$ is a small number to avoid division by zero. As stated in [6], F is precisely the recognition performance of the nearest neighbor classifier when we let $\beta \to \infty$. Since $F(U)$ depends on the distance between images, we restrict U to be an orthonormal basis. In addition, $F(U)$ does not depend on the choice of basis but on the subspace; in other words,

$$F(U) = F(UO), \text{where} O \in SO(r) \tag{4}$$

Here the underlying solution space is the Grassmann manifold $\mathcal{G}_{m,r}$. Now, learning the optimal linear subspaces becomes an optimization problem,

$$\hat{U} = \arg \max_{U \in \mathcal{G}_{m,r}} F(U) \tag{5}$$

In [6], an optimization algorithm utilizing the geometric properties of the manifold is presented. Specifically, a Markov chain Monte Carlo (MCMC) type stochastic gradient-based algorithm is used to find an optimal subspace $\hat{U}$[1]. At each iteration, the gradient vector of F with respect to U, which is a skew-symmetric matrix, is computed. By following the gradient, a new solution is generated, which is used as a proposal and is accepted with a probability that depends on the performance improvement. If the performance of the new solution is better than the current solution, it is always accepted. Otherwise, the worse the new solution's performance, the lower the probability the solution is being accepted.

However, computational cost is typically expensive which may prevent OCA from being used in certain applications. The computational complexity C_n of each iteration of this algorithm is $C_n = O(d \times (n-d) \times k_{test} \times k_{training} \times n \times d)$. C_n is obtained by the following computation. $d \times (n-d)$ is the dimension of the gradient vector. For each dimension and for each test image, the closest images in all the classes need to be found to compute the ratio in Eqn. 2 and to compute the performance F in Eqn. 1. This gives the product $k_{test} \times k_{training}$. The term $n \times d$ comes from Eqn. 3. The overall computational complexity is $C_n \times t$ where t is the number of iterations.

3. TWO-STAGE OCA

From the above analysis, we see that the computation at each iteration depends on several factors and the complexity is $O(n^2)$ in terms of n, the size of data. For typical applications, n, which is the number of pixels in the image, is relatively large. Also when n is large, the dimension of the search space will also be large.(In the Grassmann manifold, whose dimensional is $(n-d) \times d$.) Thus the algorithm can be time consuming.

As the other factors in the computational complexity can not be avoided, we can reduce the dimension of data using several methods. The idea is to reduce the dimension of the input images first and do the stochastic OCA search on the lower dimensional data space, as the search space is reduced dramatically, the search time will be reduced greatly at each iteration. Note that we require the dimension in the first step to be larger than the final dimension to be used. After we obtain the linear models in both stages, they are then combined

[1]Note that the optimal solution may not be unique.

to be a single matrix and thus this 2-OCA does not affect the computation at the testing stage.

3.1. First stage: pre-dimension reduction

The first stage of the 2-OCA is to reduce the data dimension so that the OCA searching can be performed in a lower dimension space. There are several methods that can achieve this goal. The generally used methods are PCA, FDA, QR decomposition, or even just Random Component Analysis (RCA) [9]. Accordingly, our two-stage OCA method can be named as PCA/OCA, FDA/OCA, QR/OCA and RCA/OCA; collectively, it is named 2-OCA. As PCA, FDA and RCA are commonly used, their details will not be given here. Here, we review the QR decomposition method.

In [5], Ye and Li proposed a two-stage LDA/QR method which applies QR decomposition to a small class centroid matrix in the first stage to gain the algorithm efficiency and scalability. We borrow the idea from their method and use the QR decomposition in the first stage of our QR/OCA algorithm. The first stage of our QR/OCA method maximizes the separation between different class via QR decomposition. The distinct property of the QR is low time/space complexity. Formally, it aims to compute the optimal transformation matrix G that solves the following optimization problem:

$$G = \arg\min \max_{G^T G = I_t} trace(G^T S_b G) \tag{6}$$

where

$$S_b = \frac{1}{\sqrt{N}} \sum_{i=1}^{k} N_i (m_i - m)(m_i - m)^T = H_b H_b^T, \tag{7}$$

Here m_i is the centroid of the ith class, m is the global centroid of the training data set, and H_b is given by

$$H_b = \frac{1}{\sqrt{N}} [\sqrt{N_1}(m_1 - m), \ldots, \sqrt{N_k}(m_k - m)], \tag{8}$$

The solution to eq.(6) can be obtained through a rank revealing factorization of H_b which is related to S_b in such a way that we can get G when t is the rank of S_b.

The pseudocode for QR computation is shown in Fig. 1, which is also called pre-LDA/QR algorithm in [5] as it does not use the within-class information of data. Note that the rank t of the matrix H_b is bounded from above by $k - 1$. In practice, the k centroid in the data set are usually linearly independent. In this case, the number of retained dimensions is $t = k - 1$.

3.2. Second stage: OCA search in the low dimension space

The computational costs for OCA on two Grassmann manifolds $G_{n_0,d}$ and $G_{n_1,d}$ where $n_1 = n_0/m$ and $n_0 >> d$ are easily compared. For each iteration, the computational complexity with images of size n_0 is $C_{n_0} = O(d \times (n_0 - d) \times k_{test} \times k_{training} \times n_0 \times d)$. While the computational complexity with images of size $n_1 = n_0/m$ is

Input: Data matrix A.
Output: Reduced data matrix A^L.
1. Construct the matrix H_b in (6).
2. Apply QR decomposition with column pivoting to H_b as $H_b = QR\prod$, where $Q \in \Re^{n\times t}$,$R \in \Re^{t\times k}$,$\prod \in \Re^{k\times k}$, $t = rank(H_b)$;
3. $G \leftarrow Q$. //optimal transformation
4. $A^L \leftarrow G^T A$. //reduced representation.

Fig. 1. Pre-LDA/QR Algorithm.

$$\begin{aligned} C_{n_1} &= O(d \times (\tfrac{n_0}{m} - d) \times k_{test} \times k_{training} \times \tfrac{n_0}{m} \times d) \\ &= \tfrac{n_0 - md}{m^2(n_0 - d)} C_{N_1} \\ &\approx \tfrac{1}{m^2} C_{n_0}, \end{aligned} \tag{9}$$

considering the fact $n_0 >> d$. Obviously it is much more efficient to learn on $\mathcal{G}_{n_1,d}$ than on $\mathcal{G}_{n_0,d}$ for the dimension of search space is reduced from $d \times (n_0 - d)$ to $d \times (n_1 - d)$. Therefore, we get the basis U of size $n_0 \times d$ with performing the time saving learning process in a smaller space.

4. EXPERIMENTAL RESULTS

We evaluate the effectiveness of the two-stage OCA algorithm on two well-known face datasets: ORL face dataset [2](40 individual, each 10 images, each with size 92×112) and PIE face dataset [3](66 person, 21 images each, each with size 100×100). First, we illustrate the accuracy and efficiency of this algorithm by using different method in the first stage of the algorithm, such as PCA/OCA, FDA/OCA, RP/OCA, QR/OCA, etc. Second, we compare the performance of this algorithm with other well-known classification algorithm, such as PCA, FDA, QR/FDA, etc. We use the K-Nearest Neighbor(KNN) algorithm as the classifier. The C program is running in a PC with 1.80GHz CPU, 1G RAM.

4.1. Different dimensional methods on the first stage

In order to evaluate the influence of the initial dimensional reduction method on the performance of the proposed method, the PCA/OCA, RCA/OCA, LDA/OCA and QR/OCA forms of 2-OCA were evaluated. Table 1 shows the performance of different methods on ORL and PIE datasets. For ORL face data set, we select 5 images for training and other 5 for testing. For PIE face data, we select 10 images for training and other 11 for testing. We can see our proposed method achieves high accuracy with all of the initial dimension reduction strategies.

[2]http://www.uk.research.att.com/facedatabase.html
[3]http://www.ri.cmu.edu/projects

Table 1. Classification accuracy (%) of proposed methods on the ORL and PIE data sets with different dimensional reduction in the first stage.

Data	KNN	PCA/OCA	RCA/OCA	LDA/OCA	QR/OCA
ORL	1	100	97.5	100	100
	3	100	95.0	100	100
	4	100	90.0	100	100
	5	100	90.0	100	100
	10	100	87.5	100	100
PIE	1	99.86	98.62	100	100
	3	100	93.66	100	100
	4	100	92.28	100	100
	5	98.21	90.91	100	100
	10	100	92.70	100	100

Table 2. Classification accuracy (%) of different methods on ORL data set.

KNN	PCA	PCA+LDA	QR/LDA	OCA	PCA/OCA
1	97.25	95.00	98.25	100	100
3	94.50	94.75	98.00	100	100
5	92.25	95.50	98.25	100	100
10	81.25	93.75	96.75	100	100

4.2. Comparison with other methods

In the second experiment, we compare our proposed two-stage OCA method with the other well-known classification methods, such as PCA, LDA, PCA+LDA etc. The classification accuracy is estimated using 10-fold cross-validation. Table 2 shows the performance of these methods. For this experiment, we use the PCA/OCA version of the proposed two-stage OCA algorithm. As the table shows, our method gives the best performance, which is the same as the orignal OCA but with significantly less time as shown below.

4.3. Efficiency

In this experiment, we study the computational efficiency gain of the proposed two-stage OCA algorithm by comparing its running time with that of the original OCA algorithm. Here we use the ORL data set as an example, when we do the search on the original space, the running time is about 2 days (1000 iteration), while it only takes 989 seconds when we use the PCA/OCA method (In the first stage, we reduce the dimension from 10304 to 100) with the recognition performance remains the same. We have observed similar gain in computation time on PIE and other datasets. These experimental results show clearly that the two-stage OCA algorithm speeds up the original OCA algorithm dramatically.

5. CONCLUSION

Optimal component analysis (OCA) provides a general formulation and gives an optimal solution for data classification applications. A practical limitation of OCA is that the running time of OCA algorithm is large as it uses stochastic optimization. In this paper, we have proposed a two-stage OCA algorithm which first decreases the dimension and then applied OCA on the reduced dimensional space. Experimental results demonstrate that the proposed algorithm reduces the computation time very significantly, typically by two orders of magnitude while maintaining high recognition performance.

ACKNOWLEDGEMENT

The authors would like to thank the reviews for their insightful comments and the producers of the ORL and PIE data sets for making them public. This research was supported in part by NSF grants CCF-0514743, IIS-0307998, ACI-0324944 and ARO grant W911NF-04-01-0268.

6. REFERENCES

[1] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuro-science*, vol. 3, pp. 71–86, 1991.

[2] R. O. Duda, P. E. Hart, and D. Stock, *Pattern Classification*, Wiley, 2000.

[3] A. Hyvarinen, J. Karhunen, and E. Oja, *Indenpendent Component Analysis*, John Wiley and Son, 2001.

[4] P. N. Belhumeour, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," *IEEE Transacation on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[5] J. Ye and Q. Li, "A Two-Stage Linear Discriminant Analysis via QR-Decomposition," *IEEE Transacation on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 929–941, 2005.

[6] X. Liu, A. Srivastava, and K. Gallivan, "Optimal Linear Representation of Images for Object Recognition," *IEEE Transacation on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 662–666, 2004.

[7] A. Srivastava and Xiuwen Liu, "Tools for Application-Driven Dimensional Reduction," *Neuro Computation*, vol. 67, pp. 136–160, 2005.

[8] X. Liu, A. Srivastava, and D. L. Wang, "Intrisic Generalization Analysis for Low Demensional Representation," *Neural Networks*, vol. 16, no. 5/6, pp. 537–545, 2003.

[9] Santosh S. Vempala, *The Random Projection Method*, vol. 65, American Mathematics Society, 2004.