# TWO DIMENSIONAL MAXIMUM MARGIN CRITERION

*Quanquan Gu and Jie Zhou*

State Key Laboratory on Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology(TNList)
Department of Automation, Tsinghua University, Beijing 100084, China
gqq03@mails.tsinghua.edu.cn, jzhou@tsinghua.edu.cn

## ABSTRACT

Maximum Margin Criterion is a well-known method for feature extraction and dimensionality reduction. In this paper, we propose a novel feature extraction method, namely Two Dimensional Maximum Margin Criterion (2DMMC), specifically for matrix representation data, e.g. images. 2DMMC aims to find two orthogonal projection matrices to project the original matrices to a low dimensional matrix subspace, in which a sample is close to those in the same class but far from those in different classes. Both theoretical analysis and experiments on benchmark face recognition data sets illustrate that the proposed method is very effective and efficient.

***Index Terms***— Maximum Margin Criterion, Two Dimensional, Feature Extraction

## 1. INTRODUCTION

Feature extraction is an important topic in machine learning. The most popular unsupervised feature extraction method is principal component analysis (PCA). It aims to find a subspace in which the variance of the projected data is maximum. Since PCA does not take into account the class information, the features extracted are not very suitable for classification.

Linear discriminant analysis (LDA) [1] is a supervised method which has been shown to be more effective than PCA [2]. It is based on *Fisher Criterion*, which aims to maximize the between class distance and minimize the within class distance, i.e.

$$\max \mathrm{tr}((\mathbf{W}^T\mathbf{S}_w\mathbf{W})^{-1}(\mathbf{W}^T\mathbf{S}_b\mathbf{W})), \tag{1}$$

where $\mathbf{S}_b = \sum_{i=1}^{c} n_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T$ is called between-class scatter matrix, $\mathbf{m}_i$ and $n_i$ are mean vector and size of class $i$ respectively, $\mathbf{m} = \frac{1}{n}\sum_{i=1}^{c} n_i\mathbf{m}_i$ is the overall mean vector, $\mathbf{S}_w = \sum_{i=1}^{c} \mathbf{S}_i$ is the within-class scatter matrix, $\mathbf{S}_i$ is the covariance matrix of class $i$. However, LDA suffers several drawbacks: (1) *small sample size* (SSS) problem: when the size of the dataset is small, the within-class scatter matrix $\mathbf{S}_w$ will be singular which makes the generalized eigenproblem cannot be solved; (2) it can only extract at most $c-1$ features where $c$ is the number of classes.

The essence of these drawbacks of LDA mentioned above owes to *Fisher Criterion* in Eq.(1), a promising alternative is *Maximum Margin Criterion* [3], i.e.

$$\max \mathrm{tr}(\mathbf{W}^T(\mathbf{S}_b - \lambda\mathbf{S}_w)\mathbf{W}), \tag{2}$$

where $\mathrm{tr}(\cdot)$ denotes the matrix trace and $\mathbf{W}^T\mathbf{W} = \mathbf{I}$, $\lambda$ is a weighted parameter. MMC aims to find a subspace in which a sample is close to those in the same class but far from those in different classes. It should be noted that in the original MMC [3], $\lambda$ is set to 1. MMC does not have the problem which LDA has. So MMC is usually a more efficient and robust feature extraction method.

The methods discussed above are all based on vector data. However, many real world data, e.g. image, is usually represented by matrix. When the image matrix is transformed into a vector, the image is usually represented in a very high dimensional feature space, which results in the *curse of dimensionality*. Furthermore, when we transform the image into a vector, the intrinsic spatial structure is lost. To overcome these problems, [4] proposed a generalized low-rank approximation of matrices (GLRAM) which can be seen as two dimensional PCA (2DPCA). And [5] proposed a two dimensional LDA (2DLDA), which can implicitly resolve the SSS problem suffered by LDA. These 2D methods are more computationally efficient than their 1D counterparts respectively. Furthermore, due to preserving the intrinsic spatial information of data matrix, both GLRAM and 2DLDA are evaluated empirically to be more effective than PCA and LDA respectively [4] [5].

In this paper, we propose Two Dimensional Maximum Margin Criterion (2DMMC), specifically for matrix representation data, e.g. image. 2DMMC aims to find two orthogonal projection matrices to project the original matrices to a low dimensional matrix subspace, in which a sample is close to those in the same class but far from those in different classes. In contrast to 2DLDA, the main advantage of 2DMMC is that

This work was supported by Natural Science Foundation of China under grant 60673106 and 60573062.

its convergence is rigorously guaranteed, and empirical study shows that it converges very fast, which makes the algorithm very stable and efficient. In addition, 2DMMC owns all the good properties that MMC has. As a result, 2DMMC usually outperforms 2DLDA. Both theoretical analysis and experiments on benchmark face recognition data sets illustrate that the proposed method is very effective and efficient.

The remainder of this paper is organized as follows. In Section 2 we will propose two dimensional Maximum Margin Criterion. The experiments on standard face recognition data sets are demonstrated in Section 3. Finally, we draw a conclusion in Section 4.

## 2. THE PROPOSED METHOD

### 2.1. 2D Maximum Margin Criterion

In 2DMMC, we consider data with matrix representation. Let $\mathbf{X}_i \in \mathbb{R}^{r\times c}, i = 1, 2, \ldots, n$, be the $n$ images in the dataset belonging to $\{\mathcal{C}_j\}_{j=1}^{c}$. 2DMMC aims to find two orthogonal transformation matrices $\mathbf{U} \in \mathbb{R}^{r\times l_1}$ and $\mathbf{V} \in \mathbb{R}^{c\times l_2}$, that map each $\mathbf{X}_i$ to $\mathbf{Y}_i \in \mathbb{R}^{l_1\times l_2}$, such that $\mathbf{Y}_i = \mathbf{U}^T\mathbf{X}_i\mathbf{V}, \mathbf{U}^T\mathbf{U} = \mathbf{I}, \mathbf{V}^T\mathbf{V} = \mathbf{I}$.

A natural similarity metric between matrices is the Frobenius norm. Under this metric, the within class and between class scatter matrices in vector space can be generalized to matrix space

$$\begin{aligned}\mathbf{S}_w &= \sum_{i=1}^{c}\sum_{\mathbf{X}\in\mathcal{C}_i} ||\mathbf{X}-\mathbf{M}_i||_F^2,\\ \mathbf{S}_b &= \sum_{i=1}^{c} n_i||\mathbf{M}_i-\mathbf{M}||_F^2,\end{aligned} \tag{3}$$

where $\mathbf{M}_i = \sum_{\mathbf{X}\in\mathcal{C}_i}\mathbf{X}$ and $n_i$ are the mean matrix and size of class $i$ respectively, and $\mathbf{M} = \frac{1}{n}\sum_{i=1}^{c} n_i\mathbf{M}_i$ is the overall mean matrix.

By $||\mathbf{A}||_F^2 = \text{tr}(\mathbf{A}\mathbf{A}^T)$, we can rewrite Eq.(3) as

$$\begin{aligned}\mathbf{S}_w &= \text{tr}(\sum_{i=1}^{c}\sum_{\mathbf{X}\in\mathcal{C}_i}(\mathbf{X}-\mathbf{M}_i)(\mathbf{X}-\mathbf{M}_i)^T)\\ \mathbf{S}_b &= \text{tr}(\sum_{i=1}^{c} n_i(\mathbf{M}_i-\mathbf{M})(\mathbf{M}_i-\mathbf{M})^T)\end{aligned} \tag{4}$$

In the low dimensional space resulting from the linear transformation $\mathbf{U}$ and $\mathbf{V}$, the within class and between class scatter matrices are

$$\begin{aligned}\tilde{\mathbf{S}}_w &= \text{tr}(\sum_{i=1}^{c}\sum_{\mathbf{X}\in\mathcal{C}_i}\mathbf{U}^T(\mathbf{X}-\mathbf{M}_i)\mathbf{V}\mathbf{V}^T(\mathbf{X}-\mathbf{M}_i)^T\mathbf{U})\\ \tilde{\mathbf{S}}_b &= \text{tr}(\sum_{i=1}^{c} n_i\mathbf{U}^T(\mathbf{M}_i-\mathbf{M})\mathbf{V}\mathbf{V}^T(\mathbf{M}_i-\mathbf{M})^T\mathbf{U})\end{aligned} \tag{5}$$

By Maximum Margin Criterion, the optimal transformation is obtained by maximizing

$$\tilde{\mathbf{S}}_b - \lambda\tilde{\mathbf{S}}_w. \tag{6}$$

where $\lambda$ is a weighted parameter.

As we see, the optimization is with respect to $\mathbf{U}$ and $\mathbf{V}$. And we cannot give a closed form solution. In the following, we will present an alternating scheme to optimize the objective. In other word, we will optimize the objective with respect to $\mathbf{U}$ (or $\mathbf{V}$) when fixing $\mathbf{V}$ (or $\mathbf{U}$). This procedure repeats until convergence (in our experiment, we prescribe the maximum number of iterations).

### 2.2. Computation of U

In order to compute $\mathbf{U}$, we first fix $\mathbf{V}$, Eq.(6) can be written as

$$\begin{aligned}&\tilde{\mathbf{S}}_b - \lambda\tilde{\mathbf{S}}_w\\ =\ &\mathbf{U}^T(\mathbf{S}_b^{\mathbf{V}} - \lambda\mathbf{S}_w^{\mathbf{V}})\mathbf{U},\end{aligned} \tag{7}$$

where $\mathbf{S}_b^{\mathbf{V}} = \sum_{i=1}^{c}\sum_{\mathbf{X}\in\mathcal{C}_i}(\mathbf{X}-\mathbf{M}_i)\mathbf{V}\mathbf{V}^T(\mathbf{X}-\mathbf{M}_i)^T$ and $\mathbf{S}_w^{\mathbf{V}} = \sum_{i=1}^{c} n_i(\mathbf{M}_i-\mathbf{M})\mathbf{V}\mathbf{V}^T(\mathbf{M}_i-\mathbf{M})^T$.

For fixed $\mathbf{V}$, the optimal $\mathbf{U}$ can be computed by solving a eigen-decomposition on $\mathbf{S}_b^{\mathbf{V}} - \lambda\mathbf{S}_w^{\mathbf{V}}$, i.e. is composed of the $l_1$ eigenvectors corresponding to the largest $l_1$ eigenvalues of $\mathbf{S}_b^{\mathbf{V}} - \lambda\mathbf{S}_w^{\mathbf{V}}$.

### 2.3. Computation of V

In order to compute $\mathbf{V}$, similar with the computation of $\mathbf{U}$, we first fix $\mathbf{U}$. By the property $\text{tr}(\mathbf{A}\mathbf{A}^T) = \text{tr}(\mathbf{A}^T\mathbf{A})$, then Eq.(6) can also be written as

$$\begin{aligned}&\tilde{\mathbf{S}}_b - \lambda\tilde{\mathbf{S}}_w\\ =\ &\mathbf{V}^T(\mathbf{S}_b^{\mathbf{U}} - \lambda\mathbf{S}_w^{\mathbf{U}})\mathbf{V},\end{aligned} \tag{8}$$

where $\mathbf{S}_b^{\mathbf{U}} = \sum_{i=1}^{c}\sum_{\mathbf{X}\in\mathcal{C}_i}(\mathbf{X}-\mathbf{M}_i)\mathbf{U}\mathbf{U}^T(\mathbf{X}-\mathbf{M}_i)^T$ and $\mathbf{S}_w^{\mathbf{U}} = \sum_{i=1}^{c} n_i(\mathbf{M}_i-\mathbf{M})\mathbf{U}\mathbf{U}^T(\mathbf{M}_i-\mathbf{M})^T$.

For fixed $\mathbf{U}$, the optimal $\mathbf{V}$ can be computed by solving a eigen-decomposition on $\mathbf{S}_b^{\mathbf{U}} - \lambda\mathbf{S}_w^{\mathbf{U}}$, i.e. is composed of the $l_2$ eigenvectors corresponding to the largest $l_2$ eigenvalues of $\mathbf{S}_b^{\mathbf{U}} - \lambda\mathbf{S}_w^{\mathbf{U}}$.

We summarize the 2DMMC as in Algorithm 1

### 2.4. Convergence Analysis

We propose a theorem, which governs the convergence of 2DMMC.

**Theorem 1** Algorithm 1 monotonically increases the value of objective in Eq.(6), hence it converges.

**proof:** define $f(\mathbf{U}, \mathbf{V}) = \tilde{\mathbf{S}}_b - \lambda\tilde{\mathbf{S}}_w$,
since $f(\mathbf{U}, \mathbf{V}^{(t)}) \le f(\mathbf{U}^{(t+1)}, \mathbf{V}^{(t)}), \forall\mathbf{U} \in \mathbb{R}^{r\times l_1}$,
then

$$f(\mathbf{U}^{(t)}, \mathbf{V}^{(t)}) \le f(\mathbf{U}^{(t+1)}, \mathbf{V}^{(t)}) \tag{9}$$

**Algorithm 1** Two Dimensional Maximum Margin Criterion

**Input**:Training set $\{\mathbf{X}_i\}_{i=1}^n$, desired dimensionality $l_1, l_2$, maximum number of iterations $T$;
**Output**:$\mathbf{U} \in \mathbb{R}^{r\times l_1}, \mathbf{V} \in \mathbb{R}^{r\times l_2}$;
1.Initialize $t = 0$ and $\mathbf{V}^{(0)}$ with any orthogonal matrix;
2.**While** not convergent **and** $t \leq T$
3.Compute $\mathbf{S}_b^{\mathbf{V}} = \sum_{i=1}^c \sum_{\mathbf{X}\in\mathcal{C}_i}(\mathbf{X} - \mathbf{M}_i)\mathbf{V}^{(t)}\mathbf{V}^{(t)T}(\mathbf{X} - \mathbf{M}_i)^T$, $\mathbf{S}_w^{\mathbf{V}} = \sum_{i=1}^c n_i(\mathbf{M}_i - \mathbf{M})\mathbf{V}^{(t)}\mathbf{V}^{(t)T}(\mathbf{M}_i - \mathbf{M})^T$;
4.Compute the $l_1$ eigenvectors corresponding to the largest $l_1$ eigenvalues of $\mathbf{S}_b^{\mathbf{V}} - \lambda\mathbf{S}_w^{\mathbf{V}}$ to form $\mathbf{U}^{(t)}$;
5.Compute $\mathbf{S}_b^{\mathbf{U}} = \sum_{i=1}^c \sum_{\mathbf{X}\in\mathcal{C}_i}(\mathbf{X} - \mathbf{M}_i)\mathbf{U}^{(t)}\mathbf{U}^{(t)T}(\mathbf{X} - \mathbf{M}_i)^T$, $\mathbf{S}_w^{\mathbf{U}} = \sum_{i=1}^c n_i(\mathbf{M}_i - \mathbf{M})\mathbf{U}^{(t)}\mathbf{U}^{(t)T}(\mathbf{M}_i - \mathbf{M})^T$;
6.Compute the $l_2$ eigenvectors corresponding to the largest $l_2$ eigenvalues of $\mathbf{S}_b^{\mathbf{U}} - \lambda\mathbf{S}_w^{\mathbf{U}}$ to form $\mathbf{V}^{(t+1)}$;
7.$t = t + 1$;
8.**EndWhile**

On the other hand,
since $f(\mathbf{U}^{(t+1)}, \mathbf{V}) \leq f(\mathbf{U}^{(t+1)}, \mathbf{V}^{(t+1)}), \forall \mathbf{V} \in \mathbb{R}^{r\times l_2}$,
then

$$f(\mathbf{U}^{(t+1)}, \mathbf{V}^{(t)}) \leq f(\mathbf{U}^{(t+1)}, \mathbf{V}^{(t+1)}) \tag{10}$$

Hence by Eq.(9) and Eq.(10), we have

$$f(\mathbf{U}^{(t)}, \mathbf{V}^{(t)}) \leq f(\mathbf{U}^{(t+1)}, \mathbf{V}^{(t+1)}) \tag{11}$$

Therefore, the value of objective in Eq.(6) will monotonically increase until convergence. This completes the proof.

However, the convergence of 2DLDA is not theoretically guaranteed.

### 2.5. Computational Complexity Analysis

The computation of 2DMMC is very efficient. It can be obtained by solving two eigen-decomposition in each iteration. The matrices in the eigen-decomposition are of size $l_1 \times l_1$ or $l_2 \times l_2$ in 2DMMC, so the overall computation complexity of 2DMMC is $O(t(l_1^3 + l_2^3))$ where $t$ is the number of iterations. On the other hand, the matrix eigen-decomposition is of size $rc \times rc$ in PCA, LDA and MMC, so the overall computation complexity of PCA, LDA and MMC is $O((rc)^3))$. Since the convergence of Algorithm 1 is usually very fast, e.g. about $t = 4$ iterations in our experiment, we can see that $O(t(l_1^3 + l_2^3))$ is much smaller than $O((rc)^3))$.

## 3. EXPERIMENTS

In this section, we investigate the performance of the proposed algorithm for face recognition. We compare our method with PCA, LDA, MMC [3], GLRAM [4], and 2DLDA [5].

### 3.1. Data Sets

In our experiment, we use two standard face recognition databases which are widely used as bench mark data sets in feature extraction literature.

The ORL face database[1]. There are ten images for each of the 40 human subjects, which were taken at different times, varying the lighting, facial expressions and facial details. The original images (with 256 gray levels) have size $92 \times 112$, which are resized to $32 \times 32$ for efficiency;

The Yale face database[2]. It contains 11 gray scale images for each of the 15 individuals. The images demonstrate variations in lighting condition, facial expression and with/without glasses. In our experiment, the images were also resized to $32 \times 32$;

### 3.2. Parameter Settings

For each individual, $p = 2, 3, 4$ images were randomly selected as training samples, and the rest were used for testing. The training set was used to learn a subspace, and the recognition was performed in the subspace by Nearest Neighbor Classifier. Since the training set was randomly chosen, we repeated each experiment 20 times and calculated the average recognition accuracy. In general, the recognition rate varies with the dimensionality of the subspace. The best performance obtained as well as the corresponding dimensionality is reported.

For LDA, as in [2], we first use PCA to reduce the dimensionality to $n - c$ and then perform LDA. For MMC and 2DMMC, the parameter $\lambda$ in Eq.(6) is set as $\frac{\mathrm{tr}\mathbf{S}_b}{\mathrm{tr}\mathbf{S}_w}$ according to [6]. For GLRAM, 2DLDA, 2DMMC, we set $l_1 = l_2$ and search the grid $\{1, 2, \ldots, 20\}$, and prescribe the maximum number of iterations as 20. The best performance is reported.

### 3.3. Comparative Study on Classification Accuracy

Table 1 and Table 2 show the experimental results of all the methods on the two databases respectively, where the value in each entry represents the average recognition accuracy of 20 independent trials, and the number in brackets is the corresponding projection dimensionality.

It is clear that our method outperforms the other feature extraction methods significantly on both of the two data sets.

### 3.4. Comparative Study on Convergence

In this subsection, we investigate the convergence of 2DMMC in comparison with GLRAM and 2DLDA. Take Yale data set where $p = 2$ for example. The result are shown in Fig.1, where the horizontal axis denotes the number of iterations, and the vertical axis denotes the classification accuracy. Since the convergence of 2DMMC is rigorously guaranteed, we can

[1] http://www.cl.cam.ac.uk/Research/DTG/attarchive:pub/data
[2] http://cvc.yale.edu/projects/yalefaces/yalefaces.html

**Table 1**. Face Recognition accuracy of different algorithms on the ORL database. The number in brackets is the corresponding projection dimensionality.

| Method | 2 Train | 3 Train | 4 Train |
|---|---|---|---|
| PCA | 70.67(79) | 78.88(118) | 84.21(152) |
| LDA | 72.80(25) | 83.79(39) | 90.13(39) |
| MMC | 77.97(39) | 86.32(39) | 91.63(39) |
| GLRAM | 71.30(17×17) | 79.84(11×11) | 84.73(16×16) |
| 2DLDA | 78.13(11×11) | 86.79(16×16) | 92.08(15×15) |
| 2DMMC | 78.75(12×12) | 87.50(10×10) | 92.92(8×8) |

**Table 2**. Face Recognition accuracy of different algorithms on the Yale database. The number in brackets is the corresponding projection dimensionality.

| Method | 2 Train | 3 Train | 4 Train |
|---|---|---|---|
| PCA | 46.04(29) | 49.96(44) | 55.67(58) |
| LDA | 42.81(11) | 60.33(14) | 68.10(13) |
| MMC | 52.37(14) | 61.83(14) | 67.95(15) |
| GLRAM | 49.33(6×6) | 54.17(6×6) | 57.76(5×5) |
| 2DLDA | 44.37(7×7) | 59.71(5×5) | 68.71(5×5) |
| 2DMMC | 54.37(6×6) | 63.50(9×9) | 68.86(15×15) |

see that the classification accuracy gets stable very fast. In contrast, since the convergence of 2DLDA is not guaranteed, the classification accuracy is fluctuated and not stable. As a result, 2DMMC usually needs only several iterations, which makes it very efficient (See Section 3.5). It should be noted that GLRAM is also guaranteed to converge [4], so its classification accuracy also gets stable after only several iterations.

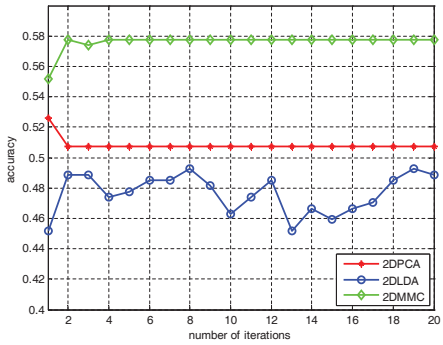


**Fig. 1**. Face Recognition accuracy with respect to the number of iterations on the Yale database with p=2 training samples.

### 3.5. Comparative Study on Efficiency

In this subsection, we investigate the computational efficiency of all the methods. We take ORL and Yale data sets where $p = 2$ for example. The result are shown in Fig.2. We can see that MMC outperforms PCA and LDA in classification accuracy (see Section 3.3) at the expense of high computational cost, while 2DMMC not only outperforms all the other methods, but also is very computational efficient. In addition, 2DMMC and GLRAM are more efficient than 2DLDA due to their fast convergence (see Section 3.4) while 2DLDA usually will not get stable until the maximum number of iterations.

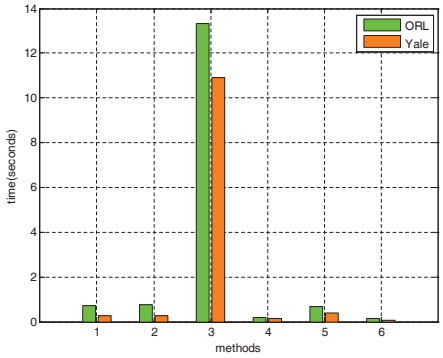


**Fig. 2**. Training time of all the methods on ORL and Yale data sets with p=2 training samples.

## 4. CONCLUSIONS

In this paper, we propose a novel Maximum Margin Criterion method, namely Two Dimensional Maximum Margin Criterion (2DMMC), specifically for matrix representation data, e.g. images. Both theoretical analysis and experiments on benchmark face recognition data sets illustrate that the proposed method is very effective and efficient.

## 5. REFERENCES

[1] Trevor Hastie, Robert Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning*, Springer, July 2001.

[2] Peter N. Belhumeur, João P. Hespanha, and David J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, 1997.

[3] Haifeng Li, Tao Jiang, and Keshu Zhang, "Efficient and robust feature extraction by maximum margin criterion," in *NIPS*, 2003.

[4] Jieping Ye, "Generalized low rank approximations of matrices," in *ICML*, 2004.

[5] Jieping Ye, Ravi Janardan, and Qi Li, "Two-dimensional linear discriminant analysis," in *NIPS*, 2004.

[6] Yangqiu Song Changshui Zhang Feiping Nie, Shiming Xiang, "Extracting the optimal dimensionality for discriminant analysis," in *ICASSP*, 2007.