MULTIPLE KERNEL 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 (MMC) is an efficient and robust feature extraction method, which has been proposed recently. Like other kernel methods, when MMC is extended to *Reproducing Kernel Hilbert Space* via kernel trick, its performance heavily depends on the choice of kernel. In this paper, we address the problem of learning the optimal kernel over a convex set of prescribed kernels for Kernel MMC (KMMC). We will give an equivalent graph based formulation of MMC, based on which we present Multiple Kernel Maximum Margin Criterion (MKMMC). Then we will show that MKMMC can be solved via alternative optimization schema. Experiments on benchmark image recognition data sets show that the proposed method outperforms KMMC via cross validation, as well as some state of the art methods.

Index Terms— Maximum Margin Criterion, Multiple Kernel Learning, Feature Extraction

1. INTRODUCTION

Feature extraction is an important topic in pattern recognition and computer vision. The most popular unsupervised feature extraction method is principal component analysis (PCA) [1]. It aims to find a subspace in which the variance of the projected data is maximized. Since PCA does not take into account the class information, the features extracted are not very suitable for classification.

Recently, a feature extraction method, named *Maximum Margin Criterion* (MMC) [2], has been proposed, which aims to find a subspace in which a point is close to those in the same class but far from those in different classes, i.e.

$$\begin{array}{ll} \max & \text{tr}(\mathbf{A}^T(\mathbf{S}_b - \mathbf{S}_w)\mathbf{A}), \\ \text{s.t.} & \mathbf{A}^T\mathbf{A} = \mathbf{I} \end{array} \qquad (1)$$

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, tr$(\cdot)$ denotes the matrix trace, and $\mathbf{I}$ denotes the identity matrix.

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

Like other kernel methods [3] [4], when MMC is extended to *Reproducing Kernel Hilbert Space* via kernel trick, its performance heavily depends on the choice of kernel. In this paper, motivated by the recent advance in multiple kernel learning [5] [6] [7], we address the problem of learning the optimal kernel over a convex set of prescribed kernels for Kernel MMC. First, we will give an equivalent graph based formulation of MMC. Second, we will present Multiple Kernel Maximum Margin Criterion (MKMMC) based on the graph formulation of MMC. Thirdly, we will show that MKMMC can be solved via alternative optimization schema. Experiments on benchmark image recognition data sets show that the proposed method outperforms KMMC via cross validation, as well as some state of the art methods.

The remainder of this paper is organized as follows. In Section 2, we present Multiple Kernel Maximum Margin Criterion. Experiments on benchmark data sets are demonstrated in Section 3. Finally, we draw a conclusion in Section 4.

2. THE PROPOSED METHOD

In this section, we will first give an equivalent graph formulation of MMC, followed which we will present the MKMMC. Finally, we will give the optimization algorithm for MKMMC.

2.1. Graph based Formulation of MMC

Recently, [8] presented a graph embedding framework, under which many dimensionality reduction methods can be unified. We will show that Maximum Margin Criterion (MMC) [2] can also been reformulated by graph. According to [8], the between-class scatter matrix $\mathbf{S}_b$ can be written as

$$\mathbf{S}_b \;=\; \sum_{i,j} ||\mathbf{x}_i - \mathbf{x}_j||^2 W_{ij}^b \qquad (2)$$

where $\mathbf{W}^b$ is defined as follows

$$W_{ij}^b = \begin{cases} \frac{1}{n} - \frac{1}{n_k}, & \text{if } y_i = y_j = k, \\ \frac{1}{n}, & \text{otherwise.} \end{cases} \tag{3}$$

where n_k is the number of points in the k-th class. And the within-class scatter matrix $\mathbf{S}_w$ can be written as

$$\mathbf{S}_w = \sum_{i,j} ||\mathbf{x}_i - \mathbf{x}_j||^2 W_{ij}^w \tag{4}$$

where $\mathbf{W}^w$ is defined as

$$W_{ij}^w = \begin{cases} \frac{1}{n_k}, & \text{if } y_i = y_j = k, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Therefore, $\mathbf{S}_b - \mathbf{S}_w$ can be written as

$$\begin{aligned} & \mathbf{S}_b - \mathbf{S}_w \\ = & \sum_{i,j} ||\mathbf{x}_i - \mathbf{x}_j||^2 (W_{ij}^b - W_{ij}^w) \\ = & \sum_{i,j} ||\mathbf{x}_i - \mathbf{x}_j||^2 W_{ij} \end{aligned} \tag{6}$$

where $\mathbf{W} = \mathbf{W}_b - \mathbf{W}_w$ is defined as

$$W_{ij} = \begin{cases} \frac{1}{n} - \frac{2}{n_k}, & \text{if } y_i = y_j = k, \\ \frac{1}{n}, & \text{otherwise.} \end{cases} \tag{7}$$

Substituting Eq.(6) into Eq.(1) leads to

$$\sum_{i,j} ||\mathbf{A}^T\mathbf{x}_i - \mathbf{A}^T\mathbf{x}_j||^2 W_{ij}. \tag{8}$$

Eq.(8) is an equivalent graph based formulation of Maximum Margin Criterion (MMC).

2.2. Kernel MMC

In [2], Kernel MMC was presented straightforwardly based on Eq.(1). Here we will derive Kernel MMC from Eq.(8), which alleviates the derivation of Multiple Kernel Maximum Margin Criterion.

We consider the problem in a feature space $\mathcal{F}$ induced by some nonlinear mapping $\phi : \mathbb{R}^d \to \mathcal{F}$. For a proper chosen ϕ, the inner product $\langle , \rangle$ in $\mathcal{F}$ is defined as

$$\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle = K(\mathbf{x}, \mathbf{y}), \tag{9}$$

where $K(,) : \mathbb{X} \times \mathbb{X} \longrightarrow \mathbb{R}$ is a positive semi-definite kernel function. The mostly used kernel functions include:

1. **Linear Kernel**:$K(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\mathbf{y}$;
2. **Polynomial Kernel**:$K(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x}^T\mathbf{y})^d$;
3. **Gaussian Kernel**:$K(\mathbf{x}, \mathbf{y}) = \exp(-\frac{||\mathbf{x}-\mathbf{y}||^2}{\gamma})$.

Let $\Phi = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \ldots, \phi(\mathbf{x}_n)]$ denote the data matrix in the feature space $\mathcal{F}$, then Eq.(8) can be extended to *Reproducing Kernel Hilbert Space* RKHS [4] as follows,

$$\sum_{i,j} ||\mathbf{A}^T\phi(\mathbf{x}_i) - \mathbf{A}^T\phi(\mathbf{x}_j)||^2 W_{ij}. \tag{10}$$

where $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_m]$. By the *Representor Theorem* [4], $\mathbf{a}_i$ are linear combinations of $\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n)$, hence there exist coefficients $\alpha_i^j, j = 1, 2, \ldots, n$, such that

$$\mathbf{a}_i = \sum_{j=1}^n \alpha_i^j \phi(\mathbf{x}_j) = \Phi\boldsymbol{\alpha}_i, \tag{11}$$

where $\boldsymbol{\alpha}_i = (\alpha_i^1, \alpha_i^2, \ldots, \alpha_i^n)^T$, and

$$\mathbf{A} = \Phi\mathcal{A}, \tag{12}$$

where $\mathcal{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m]$.

Submit Eq.(12) into Eq.(10), we obtain

$$\begin{aligned} & \sum_{i,j} ||\mathcal{A}^T\Phi^T\phi(\mathbf{x}_i) - \mathcal{A}^T\Phi^T\phi(\mathbf{x}_j)||^2 W_{ij} \\ = & \sum_{i,j} ||\mathcal{A}^T\mathbf{K}_i - \mathcal{A}^T\mathbf{K}_j||^2 W_{ij} \end{aligned} \tag{13}$$

where $\mathcal{A}^T\mathcal{A} = \mathbf{I}$ and $\mathbf{K}_i$ is defined as follows

$$\mathbf{K}_i = \begin{bmatrix} K(\mathbf{x}_1, \mathbf{x}_i) \\ K(\mathbf{x}_2, \mathbf{x}_i) \\ \vdots \\ K(\mathbf{x}_n, \mathbf{x}_i) \end{bmatrix} \tag{14}$$

Eq.(13) is exactly Kernel MMC, which is equivalent to that in [2].

2.3. Multiple Kernel MMC

In the setting of multiple kernel learning [5], we are given a set of kernel functions $\{K^t\}_{t=1}^p$. Then we are going to learn an optimal convex combination of kernels, which is restricted to

$$K \in \{K : \mathbb{X} \times \mathbb{X} \to \mathbb{R} | K = \sum_{t=1}^p \theta_t K^t, \sum_{t=1}^p \theta_t = 1, \theta_t \geq 0\}, \tag{15}$$

And the corresponding set of kernel matrices is

$$\mathbf{K} \in \{\mathbf{K} \in \mathbb{R}^{n\times n} | \mathbf{K} = \sum_{t=1}^p \theta_t \mathbf{K}^t, \sum_{t=1}^p \theta_t = 1, \theta_t \geq 0\}. \tag{16}$$

where $\mathbf{K}^t \in \mathbb{R}^{n\times n}$ are the kernel matrices computed by the kernel functions K^t.

Substitute Eq.(16) into Eq.(13), we obtain

$$\sum_{i,j} ||\mathcal{A}^T \sum_t \theta_t \mathbf{K}_i^t - \mathcal{A}^T \sum_t \theta_t \mathbf{K}_j^t||^2 W_{ij} \tag{17}$$

where $\mathbf{K}_i^t$ is defined as follows

$$\mathbf{K}_i^t = \begin{bmatrix} K^t(\mathbf{x}_1, \mathbf{x}_i) \\ K^t(\mathbf{x}_2, \mathbf{x}_i) \\ \vdots \\ K^t(\mathbf{x}_n, \mathbf{x}_i) \end{bmatrix} \tag{18}$$

By defining

$$\mathbf{G}_i = \begin{bmatrix} K^1(\mathbf{x}_1, \mathbf{x}_i) & \dots & K^p(\mathbf{x}_1, \mathbf{x}_i) \\ K^1(\mathbf{x}_2, \mathbf{x}_i) & \dots & K^p(\mathbf{x}_2, \mathbf{x}_i) \\ \vdots & \ddots & \vdots \\ K^1(\mathbf{x}_n, \mathbf{x}_i) & \dots & K^p(\mathbf{x}_n, \mathbf{x}_i) \end{bmatrix} \tag{19}$$

Eq.(17) can be rewritten as

$$\sum_{i,j} ||\mathcal{A}^T \mathbf{G}_i \boldsymbol{\theta} - \mathcal{A}^T \mathbf{G}_j \boldsymbol{\theta}||^2 W_{ij} \tag{20}$$

where $\boldsymbol{\theta} = [\theta_1, \dots, \theta_p]^T$.

Till now, we have presented the Multiple Kernel Maximum Margin Criterion (MKMMC), which is summarized as follows

$$\begin{aligned} \max \quad & \sum_{i,j} ||\mathcal{A}^T \mathbf{G}_i \boldsymbol{\theta} - \mathcal{A}^T \mathbf{G}_j \boldsymbol{\theta}||^2 W_{ij} \\ \text{s.t.} \quad & \mathcal{A}^T \mathcal{A} = \mathbf{I}, \sum_{t=1}^p \theta_t = 1, \theta_t \geq 0 \end{aligned} \tag{21}$$

2.4. Optimization

As we see, maximizing Eq.(21) is with respect to $\mathcal{A}$ and $\boldsymbol{\theta}$. And we cannot give a closed-form solution. In the following, we will present an alternating schema to optimize the objective. In other word, we will optimize the objective with respect to $\mathcal{A}$ (or $\boldsymbol{\theta}$) while fixing $\boldsymbol{\theta}$ (or $\mathcal{A}$). This procedure repeats until convergence.

2.4.1. *Computation of $\mathcal{A}$*

In order to compute $\mathcal{A}$, we fix $\boldsymbol{\theta}$, then optimizing Eq.(21) is equivalent to optimizing

$$\begin{aligned} & \max \quad \sum_{i,j} ||\mathcal{A}^T \mathbf{G}_i \boldsymbol{\theta} - \mathcal{A}^T \mathbf{G}_j \boldsymbol{\theta}||^2 W_{ij} \\ = & \max \quad \mathcal{A}^T \mathbf{L}_\theta \mathcal{A} \\ & \text{s.t.} \quad \mathcal{A}^T \mathcal{A} = \mathbf{I} \end{aligned} \tag{22}$$

where $\mathbf{L}_\theta = \sum_{ij} W_{ij}(\mathbf{G}_i - \mathbf{G}_j)\boldsymbol{\theta}\boldsymbol{\theta}^T(\mathbf{G}_i^T - \mathbf{G}_j^T)$. The optimal $\mathcal{A}$ of Eq.(22) is composed of the m eigenvectors corresponding to the largest m eigenvalues of $\mathbf{L}_\theta$.

2.4.2. *Computation of $\boldsymbol{\theta}$*

In order to compute $\boldsymbol{\theta}$, similar with the computation of $\mathcal{A}$, we fix $\mathcal{A}$. Then by the property $||\mathbf{B}|| = ||\mathbf{B}^T||$, optimizing Eq.(21) is equivalent to

$$\begin{aligned} & \max \quad \sum_{i,j} ||\boldsymbol{\theta}^T \mathbf{G}_i^T \mathcal{A} - \boldsymbol{\theta}^T \mathbf{G}_j^T \mathcal{A}||^2 W_{ij} \\ = & \max \quad \boldsymbol{\theta}^T \mathbf{L}_A \boldsymbol{\theta} \\ & \text{s.t.} \quad \sum_{t=1}^p \theta_t = 1, \theta_t \geq 0 \end{aligned} \tag{23}$$

where $\mathbf{L}_A = \sum_{ij} W_{ij}(\mathbf{G}_i^T - \mathbf{G}_j^T)\mathcal{A}\mathcal{A}^T(\mathbf{G}_i - \mathbf{G}_j)$. However, the optimal $\boldsymbol{\theta}$ cannot be solved by eigen-decomposition since the additional constraint $\sum_{t=1}^p \theta_t = 1, \theta_t \geq 0$. Eq.(23) is a linear constrained quadratic programming [9], which can be solved by quadprog() in Matlab.

3. EXPERIMENTS

In this section, we will investigate the performance of the proposed method. We compare MKMMC with MMC, KMMC via cross validation (KMMC$_{cv}$), as well as PCA and Kernel PCA via cross validation (KPCA$_{cv}$).

3.1. Data Sets

In our experiments, we use 3 standard image recognition databases which are widely used as benchmark data sets in feature extraction literature.

The ORL face data set[1] contains 10 images for each of the 40 human subjects, which were taken at different times, varying the lightings, facial expressions and facial details. The original images (with 256 gray levels) have size 92×112, which are resized to 32×32 for efficiency;

The Yale data set[2] 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 32×32;

The Coil20 data set[3] contains 32×32 gray scale images of 20 3D objects viewed from varying angles, at the interval of 5 degrees, resulting 72 images per object. The original images are resized to 32×32 for efficiency.

3.2. Parameter Settings

For each data set, we randomly divide it into training and testing sets. In detail, for each individual in the ORL and Yale data sets, $p = 2, 3, 4$ images were randomly selected as training samples, and the rest were used for testing, while for each individual in the Coil20 data set, $p = 2, 3, 4$ images were

[1]http://www.cl.cam.ac.uk/Research/DTG/attarchive:pub/data
[2]http://cvc.yale.edu/projects/yalefaces/yalefaces.html
[3]http://www1.cs.columbia.edu/CAVE/software/softlib/coil-20.php

randomly selected as training samples. We use the images in the training set to learn a subspace, and the recognition was performed in the subspace by Nearest Neighbor (NN) 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 average performance obtained as well as the corresponding dimensionality is reported.

We use Gaussian kernel for all the kernel methods. For KPCA and KMMC, the hyper-parameter γ in Gaussian kernel is set via the grid $\{2^{-5}\sigma_0^2, 2^{-4}\sigma_0^2, 2^{-3}\sigma_0^2, 2^{-2}\sigma_0^2, 2^{-1}\sigma_0^2, \sigma_0^2, 2\sigma_0^2, 2^2\sigma_0^2, 2^3\sigma_0^2, 2^4\sigma_0^2, 2^5\sigma_0^2\}$, where σ_0 is the mean distance between any two samples in the training set. We use leave-one-out cross validation on the training set, and the best γ is chosen for the testing. For MKMMC, we use 11 Gaussian kernels whose parameters correspond to the γ in the grid mentioned before.

3.3. Recognition Accuracy

Table 1, 2 and 3 show the experimental results of all the methods on the three 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.

Table 1. Classification Accuracy on ORL data set.

Method	2 Train	3 Train	4 Train
PCA	70.67(79)	78.88(118)	84.21(152)
KPCA$_{cv}$	70.67(80)	78.88(118)	84.17(153)
MMC	73.23(40)	82.80(40)	90.00(40)
KMMC$_{cv}$	75.62(39)	86.54(38)	91.79(39)
MKMMC	79.09(39)	88.84(40)	93.79(40)

Table 2. Classification Accuracy on Yale data set.

Method	2 Train	3 Train	4 Train
PCA	46.04(29)	49.96(44)	55.67(58)
KPCA$_{cv}$	46.04(29)	49.96(44)	55.71(58)
MMC	51.93(15)	61.13(15)	67.95(15)
KMMC$_{cv}$	52.15(14)	64.04(14)	71.62(14)
MKMMC	53.89(14)	66.83(14)	73.52(14)

It is clear that MKMMC outperforms KMMC via cross validation as well as the other methods on all the data sets.

Table 3. Classification Accuracy on Coil20 data set.

Method	4 Train	6 Train	8 Train
PCA	82.17 (18)	86.86(18)	89.09(16)
KPCA$_{cv}$	82.12 (18)	86.83(18)	89.05(16)
MMC	84.57(9)	89.32(13)	92.45(10)
KMMC$_{cv}$	84.69(8)	89.60(8)	93.14(8)
MKMMC	85.53(9)	90.18(12)	93.32(12)

4. CONCLUSION

In this paper, we address the problem of learning the optimal kernel over a convex set of prescribed kernels for Kernel MMC. First, We give another formulation of Kernel MMC, based on which we present Multiple Kernel Maximum Margin Criterion (MKMMC). Then we show that MKMMC can be solved via alternative optimization algorithm. Experiments on benchmark image recognition data sets show that the proposed method outperforms Kernel MMC via cross validation, as well as some state of the art methods.

5. REFERENCES

[1] I. T. Jolliffe, *Principal Component Analysis*, Series in Statistics. Springer Verlag, 2002.

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

[3] Bernhard Schölkopf, Alex J. Smola, and Klaus-Robert Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[4] B. Schölkopf and A.J. Smola, *Learning with Kernels*, MIT Press, Cambridge, MA, USA, 2002.

[5] Gert R. G. Lanckriet, Nello Cristianini, Peter L. Bartlett, Laurent El Ghaoui, and Michael I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[6] Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf, "Large scale multiple kernel learning," *Journal of Machine Learning Research*, vol. 7, pp. 1531–1565, 2006.

[7] Yen-Yu Lin, Tyng-Luh Liu, and Chiou-Shann Fuh, "Dimensionality reduction for data in multiple feature representations," in *NIPS*, 2008.

[8] Shuicheng Yan, Dong Xu, Benyu Zhang, HongJiang Zhang, Qiang Yang, and Stephen Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, 2007.

[9] Stephen Boyd and Lieven Vandenberghe, *Convex optimization*, Cambridge University Press, Cambridge, 2004.