Multi-Kernel Maximum Entropy Discrimination for Multi-View Learning

Guoqing Chao and Shiliang Sun [1]

Shanghai Key Laboratory of Multidimensional Information Processing, Department of Computer Science and Technology, East China Normal University, 500 Dongchuan Road, Shanghai 200241, China

Abstract. Maximum entropy discrimination (MED) is a general framework for discriminative estimation which integrates the principles of maximum entropy and maximum margin. In this paper, we propose a novel approach named multi-kernel MED (MKMED) for multi-view learning (MVL), which takes advantage of the complementary principle for MVL. Multiple kernels encode the similarities in different views. We obtain a kernel matrix by multiple kernel combination to make use of the complementary information in different views. Based on the kernel matrix obtained by multiple kernel combination, we can proceed MVL within the MED framework. The experimental results on multiple datasets demonstrate the effectiveness of the proposed MKMED. MKMED outperforms the single-view MEDs and a competing MVL mothod named SVM-2K, and is competitive with the state-of-the-art multi-view MED (MVMED) and even sometimes exceeds it.

Keywords. Multi-view learning, Maximum entropy discrimination, Multi-kernel learning

1. Introduction

In real-world applications, it is often extensive that many data have multiple feature representations [1,2,3,4]. For example, a web page can be described by words appearing on the web page itself and words underlying all links pointing to the web page from other pages. In multimedia learning, multimedia segments can be simultaneously described by their video signals and audio signals. As another example, in content-based web-image retrieval, an object can be described by its visual features from the image and at the same time by the text surrounding the image. How to take good advantage of the multiple feature representations? The research to deal with this problem is known as multi-view learning (MVL). These views or representations may be obtained from multiple feature sets or different sources. MVL is a rapidly growing direction in machine learning with well theoretical underpinnings and it has achieved great success in practice. A noteworthy fact for MVL is that when there are no natural multiple views, manually generated multiple views can still improve the performance [5]. Therefore, the application range for MVL is very wide. For a comprehensive survey on MVL, refer to [6,7]. A related concept is ensemble of classifiers, which can use single view or multiple views to make

[1] Corresponding Author. Tel.: +86-21-54345183; Fax: +86-21-54345119; Email:*slsun@cs.ecnu.edu.cn.*

a final decision. When each classifier just uses one of multiple views to ensemble, this kind of ensemble of classifiers can be considered as an implementation of MVL.

The existing successful MVL methods respect two significant principles: consensus and complementary [7]. While the consensus principle aims to maximize the agreement among multiple views, the complementary principle assumes that each view of the data contains some information not in other views. According to Xu et al. [7], representative MVL methods can be classified into three groups: 1) co-training [1,8,9,10], 2) multiple kernel learning, and 3) subspace learning [11,12,13]. But Xu et al. [7] just reviews some multiple kernel learning methods, which haven't been used in MVL. To deal with heterogeneous data, Lewis et.al [14] used different kernel functions to encode protein sequence and structure respectively. It is noted that if we consider the heterogeneous form of data as the views of MVL, learning from heterogeneous forms of data will be identical to MVL and they will possess the similar level of complexity. This paper attempts to utilize multiple kernel combination in a single framework, i.e., maximum entropy discrimination (MED) [15] to integrate complementary information in different views.

MED [15] is a general framework for discriminative estimation following the maximum entropy principle, which embodies the Bayesian integration of prior information with large margin constraints on observations. By introducing a selector variable into the discrimination function, Jebara and Jaakkola [16] employed MED for feature selection. Jebara [17,18] further extended MED to the problem of multi-task feature and kernel selection. On the theoretical side, Long and Wu [19] established a mistake bound that leads to a nearly optimal algorithm for learning disjunctions based on the maximum entropy principle. In recent years, Zhu and Xing [20] proposed an MED Markov network which combines MED and structure learning and thus possesses the advantages of maximum margin and probabilistic models. By adopting a Laplace prior, Zhu et al. [21] obtained a Laplace maximum margin Markov network which is a sparse model suitable for learning complex structures. In order to deal with MVL situation, Sun and Chao [22] proposed a method named multi-view maximum entropy discrimination (MVMED) to formulate and analyze the multi-view algorithm.

Based on MED, we introduce a new MVL method named multi-kernel MED (MKMED), which is different from the existing MVMED method. MVMED follows the consensus principle by enforcing the margins of the classifiers from different views to be the same. Our proposed MKMED is the first attempt to integrate multi-kernel learning into MED for MVL. Distinct from MVMED, MKMED abides by the complementary principle. Since different kernels may correspond to different notations of similarity and they have their specific advantages, we resort to multi-kernel learning to make an appropriate combination under the complementary principle. Herein, different kernels use inputs from different views. After making multiple kernel combination on different views, we obtain a comprehensive measurement of the similarity for MVL.

With regard to multi-kernel learning (MKL), there are a large quantity of such work [23,24,25,26]. Lanchriet [23] learns a kernel matrix via semidefinite programming techniques. Mao et al. [24] presented a probabilistic interpretation of MKL such that MED with a noninformative prior over multiple views is equivalent to the formulation of MKL, and they introduced a data-dependent prior based on an ensemble of kernel predictors instead of noninformative prior. Sonnenburg et al. [25] developed an efficient semi-infinite linear program for MKL to deal with large scale problems. Subrahmanya and Shin [26] proposed an algorithm named sparse MKL to perform kernel selection.

The reminder of this paper is organized as follows. Section 2 briefly reviews MED, MVMED and kernel combination. Section 3 introduces our proposed MKMED for MVL. Section 4 reports experiments on multiple real-world datasets and makes comparisons. Finally, we give conclusion and discuss some possible future work in Section 5.

2. Background knowledge

In this section, we will first introduce the classical MED framework, and then reviews a multi-view version of MED (MVMED). At last, we give a brief overview on kernel function and kernel combination.

2.1. MED

MED is similar to Bayesian learning since the posterior of model parameters requires to be inferred. Moreover, it also integrates the maximum entropy and maximum margin principles and may not need the formulation of generative distributions for data.

Suppose we have a data set $\{X_t, y_t\}$, $t = 1, ..., N$, where X_t and y_t indicate the tth input and its corresponding output $y_t \in \{\pm 1\}$. Given two class-conditional probability distributions over the examples, i.e., $p(X_t|\theta_{y_t})$ with parameters θ_{y_t}, the decision rule follows the sign of the discriminant function

$$L(X_t|\Theta) = \log \frac{p(X_t|\theta_1)}{p(X_t|\theta_{-1})} + b, \tag{1}$$

where $\Theta = \{\theta_1, \theta_{-1}, b\}$ include the model parameters and b is a bias term that can be expressed as a log-ratio of class priors $b = \log\left(p_+/(1-p_+)\right)$ with p_+ being the prior of the positive class. Alternatively, the discriminant function can be directly described by a parametric formulation without any reference to any probability model, i.e., $L(X_t|\Theta) = \boldsymbol{\theta}^{\mathrm{T}} X_t + b$ where $\Theta = \{\boldsymbol{\theta}, b\}$. Generally, MED is formulated as follows:

$$\begin{cases} \min_{p(\Theta,\boldsymbol{\gamma})} \mathrm{KL}\big(p(\Theta,\boldsymbol{\gamma}) \parallel p_0(\Theta,\boldsymbol{\gamma})\big) \\ \text{s.t.} \int p(\Theta,\boldsymbol{\gamma})[y_t L(X_t|\Theta) - \boldsymbol{\gamma}_t] d\Theta d\boldsymbol{\gamma} \geq 0 \\ \quad 1 \leq t \leq N, \end{cases} \tag{2}$$

where $\boldsymbol{\gamma} = \{\gamma_1, \ldots, \gamma_N\}$ specify the desired classification margins which reflect the maximum margin principle as in support vector machines (SVMs), $p_0(\Theta,\boldsymbol{\gamma})$ is the predefined prior distribution that the solution tends to approach, and $\mathrm{KL}\big(p(\Theta,\boldsymbol{\gamma}) \parallel p_0(\Theta,\boldsymbol{\gamma})\big)$ is the Kullback-Leibler (KL) divergence to measure the distance between $p(\Theta,\boldsymbol{\gamma})$ and $p_0(\Theta,\boldsymbol{\gamma})$. Here, instead of seeking a single parameter estimation, MED considers a more general problem of finding a distribution $p(\Theta,\boldsymbol{\gamma})$ over the parameters and margins, from which we can get the parameter distribution $p(\Theta)$ by marginalization. Correspondingly, it uses a convex combination of discriminant functions, i.e., $\int p(\Theta) L(X_t|\Theta) d\Theta$ to make model averaging rather than a single discriminant function for decisions. In addition, the solution to MED is unique as long as it exists since the optimization problem in (2) is convex with respect to $p(\Theta,\boldsymbol{\gamma})$ [15].

To solve this MED problem, we rely on the following theorem [15].

Theorem 1 *The solution to the MED problem has the general form*

$$p(\boldsymbol{\Theta},\boldsymbol{\gamma}) = \frac{1}{Z(\boldsymbol{\lambda})} p_0(\boldsymbol{\Theta},\boldsymbol{\gamma}) e^{\sum_{t=1}^{N} \lambda_t [y_t L(X_t|\boldsymbol{\Theta}) - \gamma_t]}, \tag{3}$$

where $Z(\boldsymbol{\lambda})$ is the normalization constant (partition function) and $\boldsymbol{\lambda} = \{\lambda_1, ..., \lambda_N\}$ define a set of non-negative Lagrange multipliers, one for each classification constraint. $\boldsymbol{\lambda}$ are set by finding the unique maximum of the jointly concave objective function

$$J(\boldsymbol{\lambda}) = -logZ(\boldsymbol{\lambda}). \tag{4}$$

Whether the solution to MED can be found depends entirely on whether the partition function $Z(\boldsymbol{\lambda})$ can be evaluated in a closed form, which is given as

$$Z(\boldsymbol{\lambda}) = \int p_0(\boldsymbol{\Theta},\boldsymbol{\gamma}) e^{\sum_{t=1}^{N} \lambda_t [y_t L(X_t|\boldsymbol{\Theta}) - \gamma_t]} d\boldsymbol{\Theta} d\boldsymbol{\gamma}. \tag{5}$$

After the Lagrange multipliers $\boldsymbol{\lambda}$ are obtained, the following formula is used to predict the label of a new example X

$$\hat{y} = \text{sign}\big(\mathbb{E}_{p(\boldsymbol{\Theta})}[L(X|\boldsymbol{\Theta})]\big). \tag{6}$$

2.2. MVMED

Built on MED, Sun and Chao [22] proposed an MVMED approach to make use of multiple views in a fashion named margin consistency. They enforced the margins from two views to be equal, which means that the classification confidences from different views are deemed to match each other. By this means, MVMED followed the consensus principle in MVL.

Given the multi-view dataset $\{X_t^1, X_t^2, y_t\}$, $t = 1, ..., N$, where X_t^1 and X_t^2 denote the views from view 1 and view 2, respectively, and $y_t \in \{\pm 1\}$ is the corresponding label. MVMED considers a joint probability distribution over the two view classifier parameters $\boldsymbol{\Theta}_1$, $\boldsymbol{\Theta}_2$ and the margin $\boldsymbol{\gamma}$ where $\boldsymbol{\Theta}_1 = \{\boldsymbol{\theta}_1, b_1\}$, $\boldsymbol{\Theta}_2 = \{\boldsymbol{\theta}_2, b_2\}$, and the common margin vector $\boldsymbol{\gamma} = \{\gamma_1, \ldots, \gamma_N\}$. Enforcing the large margin constraints on two views, the MVMED is formulated as

$$\begin{cases} \min_{p(\boldsymbol{\Theta}_1,\boldsymbol{\Theta}_2,\boldsymbol{\gamma})} \text{KL}\big(p(\boldsymbol{\Theta}_1,\boldsymbol{\Theta}_2,\boldsymbol{\gamma}) \parallel p_0(\boldsymbol{\Theta}_1,\boldsymbol{\Theta}_2,\boldsymbol{\gamma})\big) \\ \text{s.t.} \int p(\boldsymbol{\Theta}_1,\boldsymbol{\Theta}_2,\boldsymbol{\gamma})[y_t L_1(X_t^1|\boldsymbol{\Theta}_1) - \gamma_t] d\boldsymbol{\Theta}_1 d\boldsymbol{\Theta}_2 d\boldsymbol{\gamma} \geq 0 \\ \quad \int p(\boldsymbol{\Theta}_1,\boldsymbol{\Theta}_2,\boldsymbol{\gamma})[y_t L_2(X_t^2|\boldsymbol{\Theta}_2) - \gamma_t] d\boldsymbol{\Theta}_1 d\boldsymbol{\Theta}_2 d\boldsymbol{\gamma} \geq 0 \\ \quad 1 \leq t \leq N, \end{cases} \tag{7}$$

where $L_1(X_t^1|\boldsymbol{\Theta}_1)$ and $L_2(X_t^2|\boldsymbol{\Theta}_2)$ are discriminant functions from view 1 and view 2, respectively. $p_0(\boldsymbol{\Theta}_1,\boldsymbol{\Theta}_2,\boldsymbol{\gamma})$ is the predefined prior distribution, and $\text{KL}\big(p(\boldsymbol{\Theta}_1,\boldsymbol{\Theta}_2,\boldsymbol{\gamma}) \parallel p_0(\boldsymbol{\Theta}_1,\boldsymbol{\Theta}_2,\boldsymbol{\gamma})\big)$ is the KL divergence to measure the distance between $p(\boldsymbol{\Theta}_1,\boldsymbol{\Theta}_2,\boldsymbol{\gamma})$ and

$p_0(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\gamma})$. They may take linear formulations such as $L_1(X_t^1|\boldsymbol{\Theta}_1) = \boldsymbol{\theta}_1^{\mathrm{T}} X_t^1 + b_1$ and $L_2(X_t^2|\boldsymbol{\Theta}_2) = \boldsymbol{\theta}_2^{\mathrm{T}} X_t^2 + b_2$.

The solution to MVMED is identified by the following expression

$$\begin{aligned} p(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\gamma}) &= \frac{1}{Z(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2)} p_0(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\gamma}) \\ &\exp\big(\sum_{t=1}^{N} \lambda_{1t}[y_t L_1(X_t^1|\boldsymbol{\Theta}_1) - \gamma_t] + \sum_{t=1}^{N} \lambda_{2t}[y_t L_2(X_t^2|\boldsymbol{\Theta}_2) - \gamma_t]\big), \end{aligned} \tag{8}$$

where $Z(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2)$ is the normalization constant and $\boldsymbol{\lambda}_1 = \{\lambda_{11}, ..., \lambda_{1N}\}$, $\boldsymbol{\lambda}_2 = \{\lambda_{21}, ..., \lambda_{2N}\}$ define two sets of non-negative Lagrange multipliers, which are set by finding the unique maximum of the jointly concave objective function

$$J(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2) = -\log Z(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2). \tag{9}$$

After $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ are obtained, the following two formulae are used to predict the label of a new example (X^1, X^2) from view 1 and view 2, respectively

$$\hat{y}_1 = \mathrm{sign}\Big(\int p(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2) L_1(X^1|\boldsymbol{\Theta}_1) d\boldsymbol{\Theta}_1 d\boldsymbol{\Theta}_2\Big), \tag{10}$$

$$\hat{y}_2 = \mathrm{sign}\Big(\int p(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2) L_2(X^2|\boldsymbol{\Theta}_2) d\boldsymbol{\Theta}_1 d\boldsymbol{\Theta}_2\Big). \tag{11}$$

The prediction for a new example can also be made by using the two views together

$$\hat{y} = \mathrm{sign}\Big(\frac{1}{2}\int p(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2)\big(L_1(X^1|\boldsymbol{\Theta}_1) + L_2(X^2|\boldsymbol{\Theta}_2)\big) d\boldsymbol{\Theta}_1 d\boldsymbol{\Theta}_2\Big). \tag{12}$$

2.3. Kernel function and kernel combination

Generally, a kernel function (also known as kernel) $\kappa(\boldsymbol{x}, \boldsymbol{y})$ defines the similarity between a given pair of objects $(\boldsymbol{x}, \boldsymbol{y})$. A larger value of $\kappa(\boldsymbol{x}, \boldsymbol{y})$ indicates that $\boldsymbol{x}$ and $\boldsymbol{y}$ are similar and a smaller value indicates they are dissimilar. The kernel function must be symmetric and positive semidefinite. Given n data points with length d, we can compute the similarity between all pairs of objects in the data set. Denoted by $\boldsymbol{K}_{n\times n}$, it is called the kernel matrix. To some extent, the kernel matrix can be a sufficient representation of the raw data. The canonical kernel function is the dot products $\kappa(\boldsymbol{x}, \boldsymbol{y}) = \sum_i x_i y_i$, which is also known as linear kernel. Some other common kernel functions are polynomial kernel, radial basis function (RBF) kernel. To 'kernelize' an algorithm, we can simply replace dot products with kernel function κ.

Kernels are useful because they often make linear classifiers effective in the dataset that was previously non-separable. Kernels may encode prior knowledge about the data and similarities among non-vector and heterogeneous datasets. In addition, since different kernels can correspond to various notions of similarity of inputs coming from dif-

ferent sources or modalities, kernel combination is a possible way to integrate multiple information sources and obtain a better solution.

Many mathematical operations are closed under positive semidefiniteness, which may ensure the combined kernel is a valid kernel [31]. The most common such operation is addition: if κ_1 and κ_2 are both kernel functions, then $\kappa(\boldsymbol{x},\boldsymbol{y}) = \kappa_1(\boldsymbol{x},\boldsymbol{y}) + \kappa_2(\boldsymbol{x},\boldsymbol{y})$ is a valid kernel. A weighted combination of kernels $\eta_1\kappa_1(\boldsymbol{x},\boldsymbol{y}) + \eta_2\kappa_2(\boldsymbol{x},\boldsymbol{y})$ is a better choice for kernel combination with positive coefficients η_1 and η_2.

3. Multi-kernel maximum entropy discrimination for multi-view learning

As discussed before, there are two significant principles ensuring the success of MVL: consensus and complementary principles. Consensus principle [27,29] aims to maximize the agreement on multiple views, while complementary principle [5,30] intends to employ the complementary information in different views. For SVM-2K proposed in [27], the consensus principle is followed by forcing the constraint of consensus of two views, which can be formulated as $||f^1(x_i^1) - f^2(x_i^2)|| \leq \eta_i + \varepsilon$ where η_i is a variable that imposes consensus between the two views, and ε is a slack variable. Our proposed MKMED method respects the other principle: complementary principle. Different kernels encode different information in different views and correspond to different notations of similarity, and thus we can make a kernel combination to use the complementary information in different views. For simplicity, we start with two views, which is a special case of MVL and can be easily extended to multi-view case.

The kernel combination form we adopted is $\beta\kappa_1(\boldsymbol{x},\boldsymbol{y}) + (1-\beta)\kappa_2(\boldsymbol{x},\boldsymbol{y})$ with kernel function κ_1 encoding the first view of the data and κ_2 encoding the second view of the data, $0 \leq \beta \leq 1$. For the case with three views, we can use three kernels encoding them and use two parameters β and γ tradeoff their importance. For the case with more than three views, we can deal with it similarly, but the implementation is time-consuming. Therefore, it is worth further researching for the case with more than three views, and it will be our further research work.

Subsequently, we provide an instantiation of MED and integrate the kernel combination into the instantiation, which produces our MKMED. As to the discriminant function $L(X_t|\boldsymbol{\Theta})$, we use the linear form $L(X_t|\boldsymbol{\Theta}) = \boldsymbol{\theta}^{\mathrm{T}}X_t + b$. From (3), we find that the prior $p_0(\boldsymbol{\Theta},\boldsymbol{\gamma})$ plays an important role in the MED framework. Therefore, it is necessary to design specific prior forms for better instantiation. We suppose

$$p_0(\boldsymbol{\Theta},\boldsymbol{\gamma}) = p_0(\boldsymbol{\Theta})p_0(\boldsymbol{\gamma}) = p_0(\boldsymbol{\theta})p_0(b)p_0(\boldsymbol{\gamma}), \tag{13}$$

where $p_0(b)$ approaches a non-informative Gaussian prior, $p_0(\boldsymbol{\theta})$ is Gaussian distributed with mean $\mathbf{0}$ and identity covariance $\boldsymbol{I}$, and the prior over the margin constraints $\boldsymbol{\gamma}$ is assumed to be fully factorized

$$p_0(\boldsymbol{\gamma}) = \prod_{t=1}^{N} p_0(\gamma_t), \tag{14}$$

with $p_0(\gamma_t) = c\exp\big(-c(1-\gamma_t)\big)$ and $\gamma_t \leq 1$. A penalty is incurred for margins smaller than $1 - 1/c$ (the prior mean of γ_t) while vanishes otherwise. In fact, this choice of the

margin prior corresponds to the use of slack variables and additive penalties in SVM. The margin prior allows some slackness to handle the non-separable case, which is analogous to soft-margin SVMs. By making such assumptions, (5) becomes

$$
\begin{aligned}
Z(\boldsymbol{\lambda}) &= \int \mathcal{N}(\boldsymbol{\theta}|\mathbf{0},\mathbf{I})\mathcal{N}(b|\mathbf{0},\boldsymbol{\sigma}^2)\prod_{t=1}^{N} c\exp\big(-c(1-\gamma_t)\big) \\
&\quad \exp\big(\sum_{t=1}^{N}\lambda_t[y_t L(X_t|\boldsymbol{\Theta})-\gamma_t]\big)d\boldsymbol{\Theta}d\boldsymbol{\gamma} \\
&= \int \mathcal{N}(\boldsymbol{\theta}|\mathbf{0},\mathbf{I})\exp\big(\sum_{t=1}^{N}\lambda_t y_t \boldsymbol{\theta}^{\mathrm{T}}X_t\big)d\boldsymbol{\theta}\int \mathcal{N}(b|\mathbf{0},\boldsymbol{\sigma}^2)\exp\big(\sum_{t=1}^{N}\lambda_t y_t b\big)db \qquad (15)\\
&\quad \int\prod_{t=1}^{N} c\exp\big(-c(1-\gamma_t)\big)\exp\big(-\sum_{t=1}^{N}\lambda_t\gamma_t\big)d\boldsymbol{\gamma} \\
&= \exp\Big(\frac{1}{2}\sum_{t,\tau=1}^{N}\lambda_t\lambda_\tau y_t y_\tau {X_t}^{\mathrm{T}}X_\tau + \frac{\boldsymbol{\sigma}^2}{2}\big(\sum_{t=1}^{N}\lambda_t y_t\big)^2\Big)\prod_{t=1}^{N}\Big(\frac{c}{c-\lambda_t}e^{-\lambda_t}\Big).
\end{aligned}
$$

We substitute (15) into (4) to obtain

$$
\begin{aligned}
J(\boldsymbol{\lambda}) &= \sum_{t=1}^{N}\big[\lambda_t+\log(1-\frac{\lambda_t}{c})\big]-\frac{1}{2}\sum_{t,\tau=1}^{N}\lambda_t\lambda_\tau y_t y_\tau {X_t}^{\mathrm{T}}X_\tau \\
&\quad -\frac{\boldsymbol{\sigma}^2}{2}\big(\sum_{t=1}^{N}\lambda_t y_t\big)^2,
\end{aligned} \qquad (16)
$$

where $\lambda_t \geq 0, t = 1,...,N$. Since $\boldsymbol{\sigma}^2 \to \infty$ corresponds to using non-informative prior on the bias term b, the above objective function requires $\sum_{t=1}^{N}\lambda_t y_t = 0$. Thus, we get the following dual optimization problem

$$
\begin{cases}
\max\limits_{\boldsymbol{\lambda}} \sum\limits_{t=1}^{N}\Big(\lambda_t+\log\big(1-\frac{\lambda_t}{c}\big)\Big)-\frac{1}{2}\sum\limits_{t,\tau=1}^{N}\lambda_t\lambda_\tau y_t y_\tau {X_t}^{\mathrm{T}}X_\tau \\
\quad \sum\limits_{t=1}^{N}\lambda_t y_t = 0.
\end{cases} \qquad (17)
$$

For the above dual optimization problem, we replace ${X_t}^{\mathrm{T}}X_\tau$ with Mercer kernel function $\kappa(X_t,X_\tau)$ to obtain the kernel version of the instantiation of MED as (18).

$$
\begin{cases}
\max\limits_{\boldsymbol{\lambda}} \sum\limits_{t=1}^{N}\Big(\lambda_t+\log\big(1-\frac{\lambda_t}{c}\big)\Big)-\frac{1}{2}\sum\limits_{t,\tau=1}^{N}\lambda_t\lambda_\tau y_t y_\tau \kappa(X_t,X_\tau) \\
\quad \sum\limits_{t=1}^{N}\lambda_t y_t = 0.
\end{cases} \qquad (18)
$$

In order to handle MVL with MED, now we will use the kernel combination $\beta\kappa_1(X_{1t},X_{1\tau})+(1-\beta)\kappa_2(X_{2t},X_{2\tau})$ to substitute $\kappa(X_t,X_\tau)$ to obtain MKMED, $\beta\in[0,1]$. (X_{1t},X_{2t}) denote the two views of the data, $t=1,...,N$. We obtain the formulation for MKMED

$$\begin{cases} \max\limits_{\boldsymbol{\lambda}} \sum\limits_{t=1}^{N}\Big(\lambda_t+\log\big(1-\dfrac{\lambda_t}{c}\big)\Big) \\ \quad -\dfrac{1}{2}\sum\limits_{t,\tau=1}^{N}\lambda_t\lambda_\tau y_t y_\tau\big(\beta\kappa_1(X_{1t},X_{1\tau})+(1-\beta)\kappa_2(X_{2t},X_{2\tau})\big) \\ \quad \sum\limits_{t=1}^{N}\lambda_t y_t=0. \end{cases} \tag{19}$$

From the above formulation, we find that if $\beta=0$ MKMED will degenerate to MED with only the second view, and if $\beta=1$ MKMED will degenerate to MED with only the first view. In order to facilitate the expression, we will denote the above two cases by MKMED2 and MKMED1, respectively.

In order to better understand the procedure of MKMED, we give the algorithm of MKMED in Algorithm 1. Apart from the input and output, we introduce the algorithm of MKMED with three steps: preparation, training and test mainly. They are detailed in the execution part of Algorithm 1.

Algorithm 1 MKMED

Input:

Training sets $\{X_{1t},X_{2t},y_t\}$, training set size ℓ, test sets $\{X_{1i},X_{2i}\}$, test set size u, parameter c, tradeoff parameter β, kernel functions κ_1 and κ_2.

Execution:

preparation:

Initialize $\boldsymbol{\lambda}$.

Make kernel combination $\beta\kappa_1(X_{1t},X_{1\tau})+(1-\beta)\kappa_2(X_{2t},X_{2\tau})$.

Training:

Solve the optimization problem (19) with ℓ replacing N.

Test:

Once the Lagrange multipliers $\boldsymbol{\lambda}$ are obtained, use any of the formulae (10), (11) and (12) to make predictions for the test sets (X_{1i},X_{2i}), $i=\ell+1,...,\ell+u$.

Output:

The prediction outputs of the test sets and the accuracy of these predictions.

For our proposed MKMED, its computational complexity is $O(\ell^3)$ with ℓ indicating the number of the training set, which is the same with SVM. This cubic complexity is a challenge for large scale application. In fact, standard SVM also suffers this disadvantage and special optimizations needs to be designed. Therefore, it is interesting to design some speedup strategies such as stochastic optimization and the Nyström approximation to make MKMED scalable, which will be our future work.

4. Experiments

In order to validate the effectiveness of our proposed MKMED, we evaluate it on three real-world datasets: web-page, ionosphere and advertisement.

4.1. Datasets

Table 1. The average classification accuracies and standard deviations (%) of six methods on three datasets.

Dataset	MKMED1	MKMED2	MKMED3	SVM-2K	MVMED	MKMED
Web-page	84.68±2.36	92.85±1.03	92.81±1.22	90.42±2.44	92.93±2.07	**93.31**±1.26
Ionosphere	86.59±3.38	**1**±0	**1**±0	98.46±1.25	**1**±0	**1**±0
Advertisement	93.60±1.23	93.00±1.55	94.13±1.33	93.66±1.73	**95.47**±1.12	94.67±1.54

Web-page dataset: The web-page dataset[2] includes 1051 web pages collected from computer science department web sites at four U.S. universities: Cornell University, University of Washington, University of Wisconsin, and University of Texas [1]. The task is to predict whether a web page is a course home page or not. Within the dataset there are 230 course pages and 821 non-course pages. One view is words occurring in a web page while the other is words appearing in the links pointing to that page. We preprocess them into 2333 and 87 dimensions respectively. For convenience and effectiveness, the dimension of view 1 is reduced from 2333 to 500 via principle component analysis (PCA).

Ionosphere dataset: The ionosphere dataset[3] is collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. The targets are free electrons in the ionosphere. Good radar returns are those showing evidence of some type of structure in the ionosphere. Bad radar returns are those that not and their signals pass through the ionosphere. There are 225 "good" (positive) instances and 126 "bad" (negative) instances in this dataset. Originally it has only one view with 35 dimensions, but we generate the other view via PCA. The generated view has 24 dimensions.

Advertisement dataset: This dataset[4] represents a set of possible advertisements on Internet pages. The task is to predict whether an image is an advertisement ("ad") or not ("nonad"). The dataset consists of 3279 examples including 459 "ad" images and 2820 "nonad" images. The original features of the dataset consist of three continuous values and 1554 binary values. One or more of the three continuous features are missing in 28% of the instances; missing values should be interpreted as "unknown". We omit the three continuous features and divide the other features into two views. The first view describes the image itself (words in the image's URL, alt text and caption), while the second view contains all other features (words from the URLs of the pages that contain the image and the image points to). The dimensions of the two views are 587 and 967, respectively. 600 examples are sampled to be the used dataset.

[2]http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-51/www/co-training/data/

[3]http://archive.ics.uci.edu/ml/datasets/Ionosphere

[4]http://archive.ics.uci.edu/ml/datasets/Internet+Advertisements

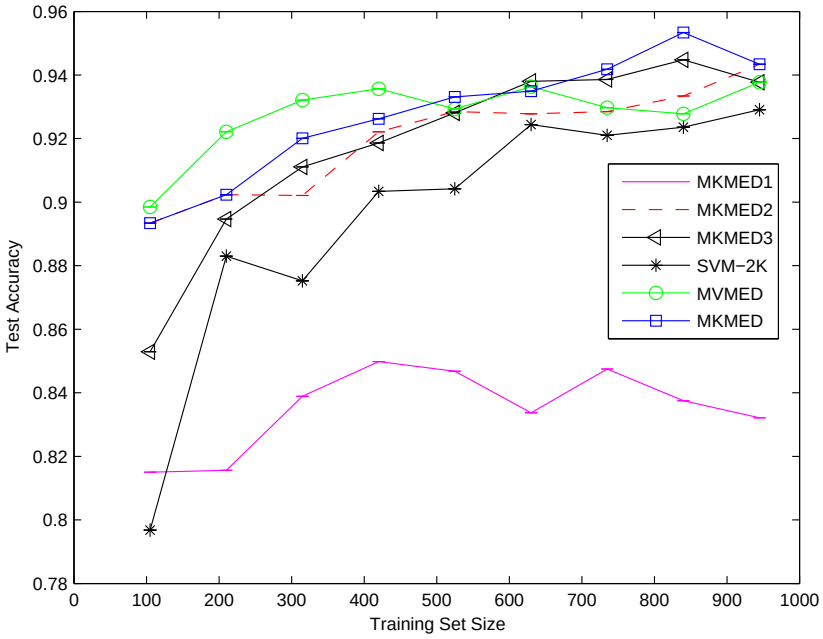

Figure 1. The performance on Web-page classification.

4.2. Setup

We compare the proposed MKMED with the following baselines:

- **MKMED1 and MKMED2**: They either only use the feature from the first view or just use the feature from the second view of each dataset. MKMED1 is the MKMED with $\beta = 1$ while MKMED2 is the MKMED with $\beta = 0$.
- **MKMED3**: It first concatenates the two views and then use the concatenated feature to train single-view MKMED. Therefore, it uses all the views together.
- **SVM-2K** [27]: It is an MVL approach which combines KCCA and SVM into a single optimization problem.
- **MVMED** [22]: It learns a classifier within MVMED framework, which is an MVL method with MED.

These baselines are classified into single-view methods and multi-view methods. The comparison with single-view methods is to show the effectiveness of the MVL of the proposed MKMED. The comparison with multi-view methods will demonstrate whether this MVL method is superior to other multi-view methods.

For all the experiments, we will divide the dataset into the training set and test set. With the training set we will train a classifier, and then select the parameters on the validation set which is half of the test set, at last we will give the results on the unseen test set which is the other half of the test set. The parameter c will be chosen in the range $\{2^{-5}, 2^{-4}, ..., 2^{5}\}$ and β is selected from $\{0, 0.1, ..., 1\}$ via a grid search strategy. The average accuracies obtained by ten random divisions of the training and test sets are

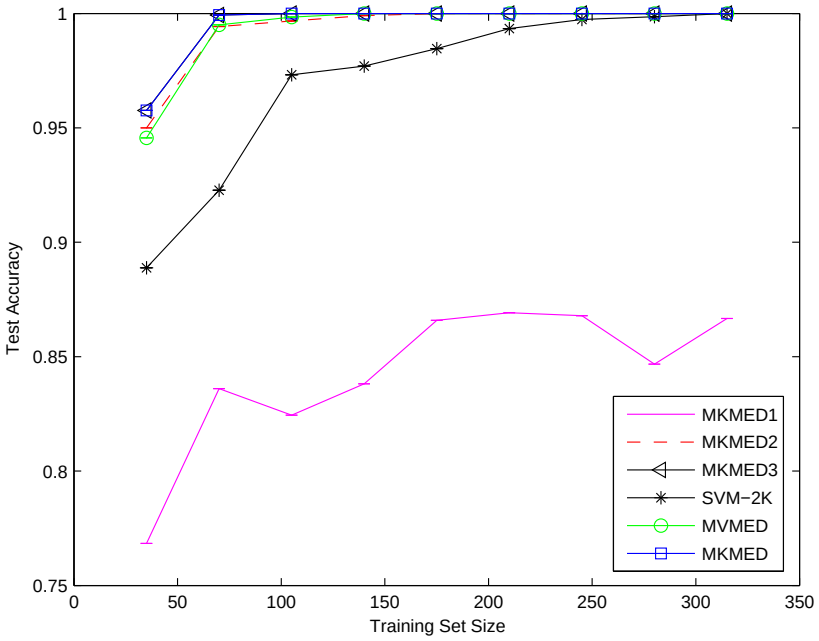

Figure 2. The performance on Ionosphere classification.

reported. For all the datasets, we choose linear kernels for both views and first report the average accuracies and standard deviations of the five methods with half of the dataset as the training set in Table 1, and then we conduct experiments with varying training set sizes and show the results in Figures 1$\sim$ 3. To make it clear, we have divided each data set into ten parts averagely, and then use 1,2,...,9 parts as the training set and the rest parts as the validation and unseen test set. The sizes of the validation set and unseen test set equal. Note that the results in Table 1 is just one case of that in Figures 1$\sim$ 3, we list it out to demonstrate the corresponding numerical values in the case where half of the dataset (five parts) is used as the training set. Clearly, the comprehensive comparison should be seen from Figures 1$\sim$ 3 combined with Table 1.

With respect to MKMED, before kernel combination, each kernel is first centered around the origin in the feature space, and each data point is projected into the unit sphere using $\hat{\kappa}(\boldsymbol{x},\boldsymbol{y}) = \kappa(\boldsymbol{x},\boldsymbol{y})/\sqrt{\kappa(\boldsymbol{x},\boldsymbol{x})\kappa(\boldsymbol{y},\boldsymbol{y})}$. Subsequently, we will show the experimental results on the three datasets.

4.3. Results

From Table 1, we find that our proposed MKMED outperforms all the other methods on the web-page dataset. From Figure 1, we see that MKMED performs worse than MVMED at first and then catches up with MVMED and performs the best. MKMED3 also performs worse than MVMED at first and then catches up with MVMED and outperforms finally, but it performs worse than MKMED all the time. MKMED, MVMED and MKMED3 perform better than other methods.

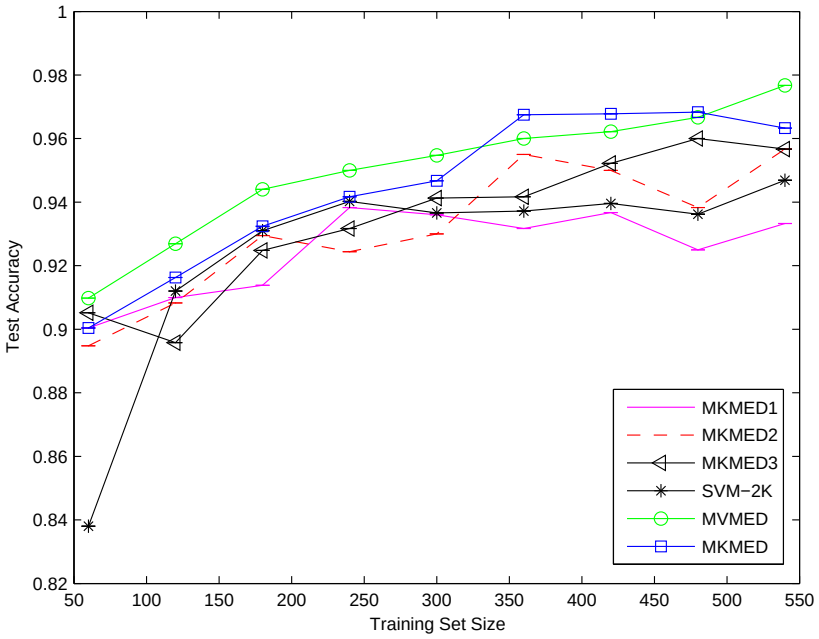

Figure 3. The performance on Advertisement classification.

Table 1 and Figure 2 both show that MKMED, MVMED, MKMED3 and MKMED2 all perform the best on the ionosphere dataset. SVM-2K performs worse when the training set is small, but with the increasing training data size, SVM-2K catches up and performs as well as MKMED, MVMED, MKMED3 and MKMED2. MKMED1 performs the worst all the time.

Table 1 demonstrates that on the advertisement dataset, MVMED performs the best and MKMED performs a little worse than MVMED but MKMED performs better than other methods. From Figure 3, we also find that MVMED and MKMED perform better than other methods, MKMED performs basically a little worse than MVMED but sometimes it performs better than MVMED. MKMED3 performs worse than MVMED and MKMED but performs better than or as well as other methods.

The experimental results on the three datasets show that MKMED performs better than three single-view learning methods MKMED1, MKMED2 and MKMED3 and a competing MVL method SVM-2K. In addition, MKMED also shows comparative performance with state-of-the-art MVMED, and sometimes even exceeds it. This not only shows that the complementary principle in MVL is effective , but only demonstrates that the proposed MKMED is competitive with MVMED in classification performance.

5. Conclusions and future work

We have proposed a novel MVL method MKMED, which takes good advantage of the complementary information in different views via MKL. First we use multiple kernel

functions encoding different views to make kernel combination, and then we integrate the kernel combination to MED for MVL. Experimental results on real-world applications web-page classification, ionosphere classification and advertisement classification demonstrate that the proposed MKMED is competitive with, and sometimes exceeds the state of the art.

For future work, it is worthy to explore integrating other kernel combination methods such as the nonlinear kernel combination [28] to the MED framework for MVL. In addition, it is also very interesting and important to investigate how to design specific speedup algorithm to deal with large-scale datasets in the future.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Project 61370175, and Shanghai Knowledge Service Platform Project (No. ZF1213).

References

[1] A. Blum and T. Mitchell, Combining labeled and unlabeled data with co-training, in: *Proceedings of the 11th Annual Conference on Computational Learning Theory* (1998), 92–100.

[2] D. R. Hardoon, S. Szedmak and J. Shawe-Taylor, Canonical correlation analysis: An overview with application to learning methods, *Neural Computation* **16** (2004), 2639–2664.

[3] B. McFee and G. Lanckriet, Learning multi-modal similarity, *Journal of Machine Learning Research* **12** (2011), 491–523.

[4] P. Xie and E. P. Xing, Multi-modal distance metric learning, in: *Proceedings of the 23th International Joint Conference on Artificial Intelligence* (2013), 1806–1812.

[5] K. Nigam and R. Ghani, Analyzing the effectiveness and applicability of co-training, in: *Proceedings of the 9th International Conference on Information and Knowledge Management* (2000), 86–93.

[6] S. Sun, A survey of multi-view machine learning, *Neural Computing and Applications* **23** (2013), 2031–2038.

[7] C. Xu, D. Tao and C. Xu, A survey on multi-view learning, CoRR **abs/1304.5634** (2013).

[8] V. Sindhwani, P. Niyogi and M. Belkin, A co-regularization approach to semi-supervised learning with multiple views, in: *Proceedings of the 22th International Conference on Machine Learning Workshop on Learning with Multiple Views* (2005), 74–79.

[9] W. Wang and Z. H. Zhou, A new analysis of co-training, in: *Proceedings of the 27th International Conference on Machine Learning* (2010), 1135–1142.

[10] S. Yu, B. Krishnapuram, R. Rosales and R. B. Rao, Bayesian co-training, *Journal of Machine Learning Research* **12** (2011), 2649–2680.

[11] N. Chen, J. Zhu, F. Sun, and E. P. Xing, Large-margin predictive latent subspace learning for multiview data analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34** (2012), 2365–2378.

[12] H. Hotelling, Relations between two sets of variants, *Biometrika* **28** (1936), 321–377.

[13] M. White, X. Zhang, D. Schuurmans and Y. Yu, Convex multi-view subspace learning, *Advances in Neural Information Processing Systems* **25** (2012), 1682–1690.

[14] D.P.Lewis, T. Jebara and W. S. Noble, Support vector machine learning from heterogeneous data: An empirical analysis using protein sequence and structure, *Bioinformatics* **22** (2006), 2753–2760.

[15] T. Jaakkola, M. Meila and T.Jebara, Maximum entropy discrimination, *Advances in Neural Information Processing Systems* **12** (1999), 470–476.

[16] T. Jebara and T. Jaakkola, Feature selection and dualities in maximum entropy discrimination, in: *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence* (2000), 291–300.

[17] T. Jebara, Multi-task feature and kernel selection for SVMs, in: *Proceedings of the 21st International Conference on Machine Learning* (2004), 55–62.

[18] T. Jebara, Multitask sparsity via maximum entropy discrimination, *Journal of Machine Learning Research* **12** (2011), 75–110.

[19] P. M.Long and X. Wu, Mistake bounds for maximum entropy discrimination, *Advances in Neural Information Processing Systems* **17** (2004), 833–840.

[20] J. Zhu and E. P. Xing, Maximum entropy discrimination Markov networks, *Journal of Machine Learning Research* **10** (2009), 2531–2569.

[21] J. Zhu, E. P. Xing and B. Zhang, Laplace maximum margin Markov networks, in: *Proceedings of the 25th International Conference on Machine Learning* (2008), 1256–1263.

[22] S. Sun and G. Chao, Multi-view maximum entropy discrimination, in: *Proceedings of the 23th International Joint Conference on Artificial Intelligence* (2013), 1706–1712.

[23] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui and M. I. Jordan, Learing the kernel matrix with semidefinite programming, *Journal of Machine Learning Research* **5** (2004), 27–72.

[24] Q. Mao, I. W. Tang, S. Gao and L. Wang, Generalized multiple kernel learning with data-dependent priors, *IEEE Transactions on Neural Networks and Learning Systems* (2014). DOI 10.1109/TNNLS.2014.2334137.

[25] S. Sonnenburg, G. Rätsch and C. Schäfer, A general and efficient multiple kernel learning algorithm, *Advances in Neural Information Processing Systems* **19** (2006), 1273–1280.

[26] N. Subrahmanya and Y. C. Shin, Sparse multiple kernel learning for signal processing applications, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32** (2010), 788–798.

[27] J. Farquhar, D. Hardoon, H. Meng, J. Shawe-Taylor and S. Szedmak, Two view learning: SVM-2K, theory and practice, *Advances in Neural Information Processing Systems* **18** (2005), 355–362.

[28] J. Li and S. Sun, Nonlinear combination of multiple kernels for support vector machines, in: *Proceedings of the 20th International Conference on Pattern Recognition* (2010), 2889–2892.

[29] S. Dasgupta, M. L. Littman and D. McAllester, Pac generalization bounds for co-training, *Advances in Neural Information Processing Systems* **25** (2012), 375–382.

[30] W. Wang and Z. H. Zhou, Analyzing co-training style algorithms, *Machine Learning: ECML*, (2007), 454–465.

[31] C. M. Bishop, Pattern recognition and machine learning, *Springer-Verlag*, 2006.