An Efficient Approach to Integrating Radius Information into Multiple Kernel Learning

Xinwang Liu, Student Member, IEEE, Lei Wang, Senior Member, IEEE, Jianping Yin, En Zhu, and Jian Zhang, Senior Member, IEEE

Abstract—Integrating radius information has been demonstrated by recent work on multiple kernel learning (MKL) as a promising way to improve kernel learning performance. Directly integrating the radius of the minimum enclosing ball (MEB) into MKL as it is, however, not only incurs significant computational overhead but also possibly adversely affects the kernel learning performance due to the notorious sensitivity of this radius to outliers. Inspired by the relationship between the radius of the MEB and the trace of total data scattering matrix, this paper proposes to incorporate the latter into MKL to improve the situation. In particular, in order to well justify the incorporation of radius information, we strictly comply with the radius–margin bound of support vector machines (SVMs) and thus focus on the \( \ell_2 \)-norm soft-margin SVM classifier. Detailed theoretical analysis is conducted to show how the proposed approach effectively preserves the merits of incorporating the radius of the MEB and how the resulting optimization is efficiently solved. Moreover, the proposed approach achieves the following advantages over its counterparts: 1) more robust in the presence of outliers or noisy training samples; 2) more computationally efficient by avoiding the quadratic optimization for computing the radius at each iteration; and 3) readily solvable by the existing off-the-shelf MKL packages. Comprehensive experiments are conducted on University of California, Irvine, protein subcellular localization, and Caltech-101 data sets, and the results well demonstrate the effectiveness and efficiency of our approach.

Index Terms—Class separability measure, enclosing minimum ball, kernel methods, multiple kernel learning (MKL), radius–margin bound, support vector machines (SVMs).

I. INTRODUCTION

A

S ONE OF the most popular machine learning approaches, kernel methods [1] have been widely used in many real-world applications such as text classification [2], [3], computer vision [4]–[6], and medical imaging analysis [7], [8], to name a few. For a given classification task, how to infer an appropriate kernel is crucial for the success of kernel methods. In the literature, many efforts have been devoted to developing kernel learning algorithms [9]–[12]. Among them, multiple kernel learning (MKL) has been considered as an effective and efficient way to achieve this goal [11], [13]–[24], where the optimal kernel is modeled as a linear combination of a set of base kernels. The idea of MKL can be generally applied to all sorts of kernel-based classifiers, such as the commonly used support vector machines (SVMs) [34] and kernel linear discriminant analysis [35], leading to SVM-based MKL and discriminant MKL, respectively. The SVM-based MKL algorithms [11], [13]–[33] iteratively learn the combination weights of base kernels and the structural parameters of SVMs, while the discriminant MKL algorithms [36]–[38] alternatively optimize the optimal projection direction and the base-kernel weights. Our work in this paper will only focus on SVM-based MKL formulations. In addition to bearing the advantage of learning the optimal kernel, another amazing property of MKL is that it provides a general framework for multisource (or multimodal) data fusion by representing each source data via a base kernel. On the other hand, the widely studied multiview learning [39]–[42] can also be related to MKL from this perspective. Altogether, these demonstrate the necessity of developing the MKL techniques and the benefits that they can bring forth.

The SVM-based MKL algorithm was first formulated and solved as a semidefinite programming problem in the pioneering work in [11]. Subsequently, it has been solved by semi-infinite linear programming [17], subgradient descent method [18], extended level method [29], and a solution with a closed form [19], which has well improved the efficiency of SVM-based MKL for practical applications [20]–[22]. Recently, two new research trends on SVM-based MKL can be observed. One focuses on developing nonsparse SVM-based MKL algorithms, which imposes an \( \ell_p \)-norm (\( p > 1 \)) constraint on the base-kernel weights in order to improve the existing SVM-based MKL dominated by the \( \ell_1 \)-norm constraint [14], [19], [30], [43]. The other focuses on employing the generalization error bound of SVMs to learn the optimal base-kernel weights [20], [31], [32]. Different from most of the existing SVM-based
MKL algorithms which maximize the margin only, these works incorporate the radius of the minimum enclosing ball (MEB) of training data into SVM-based MKL formulation by considering that the generalization error of SVMs is upper bounded by the ratio of the radius to the margin, so called the radius–margin bound [10]. Our work in this paper falls into the second research trend by proposing an efficient way to incorporating the radius information into the existing SVM-based MKL framework and analyzing its characteristics.

Several methods have recently been proposed to incorporate the radius into SVM-based MKL. As shown in [31], the ratio of the radius of the MEB to the margin is minimized to seek the optimal kernel weights. The optimization problem is then approximated by a convex one to make it efficiently solved. Nevertheless, the formulation proposed in [31] cannot handle the scaling and initialization problems in existing MKL algorithms, as pointed out in [32]. In [32], the radius of the MEB is incorporated by solving a trilevel optimization method, which adds one extra level of quadratic optimization on top of the existing SVM-based MKL framework. As shown in that work, the radius is not only an integral part of the generalization error bound of SVMs but also a critical quantity to handle the scaling and initialization issues in existing MKL algorithms. However, the method proposed in [32] has the following two drawbacks. First, compared with the existing SVM-based MKL, it leads to a more complicated learning structure, and this considerably increases the computational overhead; second, the notorious sensitivity of the radius of the MEB to outliers or noisy samples can adversely affect the kernel learning performance of MKL algorithms. These drawbacks can have a considerable impact on the efficiency and robustness of the existing MKL algorithms. Also, the work in [32] focuses on $\ell_1$-norm soft-margin formulation of SVMs, which does not tightly couple with the radius–margin bound as desired.

From our point of view, the underlying cause for the aforementioned drawbacks lies at that the radius of the MEB is incorporated into MKL as it is. To improve this situation, this paper proposes to incorporate the trace of the total scattering matrix of training data into MKL, which is inspired by its close relationship with the radius of the MEB [44]. In particular, to well justify the incorporation of radius information, we strictly comply with the radius–margin bound and focus on the $\ell_2$ norm constraint is imposed on base-kernel weights during the MKL learning process, as what were done in [11]. This further improves the computational efficiency of our approach by avoiding the time-consuming cross-validation process. Comprehensive experiments are conducted on multiple benchmark data sets, including UCI, protein subcellular localization, and Caltech-101, to compare the proposed approach with those developed in the recent literature [31], [32] with respect to classification accuracy and computational efficiency. The experimental results well demonstrate the efficiency and effectiveness of the proposed approach. In sum, this work provides an efficient way for the existing SVM-based MKL methods to utilize the information of the radius of the MEB. In addition, it is worth noting that the framework used by this paper can also be modified to improve the MKL algorithm in [32], as demonstrated in our recent work in [46]. However, that work still focuses on the $\ell_1$-norm soft-margin formulation of SVMs and does not strictly follow the radius–margin bound, as achieved in this paper.

The rest of this paper is organized as follows. We review the related work in Section II. In Section III, we give the formulation of the proposed $\ell_2$trStMKL. Then, we prove that its optimization problem can be written into the common form of existing SVM-based MKL and show how it can be efficiently solved. After that, some preliminary discussion on the proposed trace–margin criterion and radius–margin bound is conducted. Section IV reports our experimental results, and the conclusion is in Section V.

II. RELATED WORK

This section first reviews the traditional form of the MKL algorithms and then focuses on the work in [31] and [32] that incorporates the radius information into traditional MKL algorithms. Let $x_i$ be a training sample, where $x_i \in \mathbb{R}^d$ and $i = 1, \ldots, n$. The training set is denoted as $X = (x_1, \ldots, x_n) \in \mathbb{R}^{n \times d}$, where $n$ and $d$ are the number of training samples and the dimensionality of $x_i$, respectively. Let $y = (y_1, \ldots, y_n) \in \mathbb{R}^n$ be the label vector, with $y_i \in \{\pm 1\}$ being the class label of the $i$th training sample. Let $\phi_p : \mathbb{R}^d \rightarrow \mathcal{H}_p$ be the $p$th feature mapping, inducing a kernel $k_p$ in the Hilbert space $\mathcal{H}_p$, where $p = 1, \ldots, m$. $K_p$ is the kernel matrix computed with $k_p$ on the training set $X$. In the SVM-based MKL framework, each sample $x$ is mapped onto $m$ feature spaces by $\phi(x; \gamma) = [\sqrt{\gamma_1}\phi_1(x), \ldots, \sqrt{\gamma_m}\phi_m(x)]^\top$, where $\gamma$ is the weight of the $p$th base kernel. In doing so, the employed kernel can be expressed as a linear combination of $p$ base kernels, $k(\gamma) = \sum_{p=1}^{m} \gamma_p K_p$.

To seek the optimal combination weight for each base kernel, most MKL algorithms such as [17], [18], and [29] aim to
maximize the margin by solving the following optimization problem with respect to $\gamma$:  
\[
\begin{align*}
\min_{\gamma, \omega, b, \xi} & \quad \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^{n} \xi_i \\
n\text{s.t.} & \quad y_i (\langle \omega^\top \phi(x_i; \gamma) + b \rangle - 1) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \forall i \quad (1)
\end{align*}
\]
where $\omega$ is the normal of the separating hyperplane, $b$ is the bias term, and $\xi = [\xi_1, \ldots, \xi_n]^\top$ is the vector of slack variables.

Recent work on MKL has considered to incorporate the radius of the MEB into the traditional formulation and demonstrated that it helps to achieve better kernel learning performance [31, 32]. The theoretical justification for incorporating the radius lies at that the generalization error bound of SVMs is dependent on both the margin and the radius of the MEB of training data [10]. As pointed out in [32], only maximizing the margin with respect to $\gamma$ will lead to scaling and initialization issues. A larger margin can be arbitrarily achieved by scaling $\gamma$ to $\gamma \tau > 1$, and this will affect the convergency of the optimization problem. Usually, a norm constraint is imposed on $\gamma$ to address this issue. Nevertheless, identifying an appropriate norm constraint for a given kernel learning task remains an open issue [14], [33], [43], [47], [48]. Moreover, even if a norm constraint is imposed, a good kernel could still be misjudged as a poor one by simply downscaling the corresponding kernel weight [32]. These issues can be removed or mitigated by the incorporation of radius information.

The following formulation is adopted by both works in [31] and [32], with an additional $l_1$-norm constraint used in [31]
\[
\begin{align*}
\min_{\gamma, \omega, b, \xi} & \quad \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^{n} \xi_i^2 \\
n\text{s.t.} & \quad y_i (\langle \omega^\top \phi(x_i; \gamma) + b \rangle - 1) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \forall i \quad (2)
\end{align*}
\]
where $R^2$ is the squared radius of the MEB and $q = 1$ or $2$. Like the margin, $R^2$ is also a function of $\gamma$. The work in [31] focuses on how to approximate the optimization problem in (2) with one that can be more efficiently solved and does not address the scaling issue mentioned earlier. Differently, the work in [32] directly solves the optimization in (2) and carefully discusses how the scaling issue can be addressed. In detail, a trilevel optimization problem is proposed in that work
\[
\begin{align*}
\min_{\gamma} & \quad \tilde{f}(\gamma) \quad \text{s.t.} \quad \gamma_p \geq 0 \quad \forall p \\
\end{align*}
\]
where
\[
\tilde{f}(\gamma) = \left\{ \begin{array}{ll}
\max_{\alpha} & \quad \alpha^\top 1 - \frac{1}{2R^2} (\alpha \circ y)^\top K(\gamma) (\alpha \circ y) \\
n\text{s.t.} & \quad \alpha^\top y = 0 \quad 0 \leq \alpha_i \leq C \quad \forall i
\end{array} \right\} \quad (4)
\]
\[
\begin{align*}
R^2 &= \left\{ \begin{array}{ll}
\max_{\beta} & \quad \beta^\top \text{diag}(K(\gamma)) - \beta^\top K(\gamma) \beta \\
n\text{s.t.} & \quad \beta^\top 1 = 1 \quad 0 \leq \beta_i \quad \forall i
\end{array} \right\} \quad (5)
\end{align*}
\]
To solve the optimization problem, a trilevel optimization structure is developed accordingly. Specifically, in the first step, $R^2$ is computed by solving the quadratic programming (QP) in (5) with a given $\gamma$. Then, the obtained $R^2$ is taken into (4) to solve another QP to calculate $\tilde{f}(\gamma)$. The last step is to update the kernel weight $\gamma$. The aforementioned procedure is repeated until a stopping criterion is satisfied. Compared with traditional MKL algorithms, an extra QP is introduced and solved at each iteration. This can considerably increase the computation cost of SVM-based MKL, particularly when the size of the training set is large. Worse is that the solution of the optimization problem in (5) is sensitive to outliers. As a result, the obtained $R^2$ can become noisy, and this noise will, in turn, affect the optimization of kernel weights via the trilevel optimization structure. To reduce the sensitivity to outliers, we could simply impose a box constraint on $\beta$, and (5) becomes
\[
\tilde{R}^2 = \left\{ \begin{array}{ll}
\max_{\beta} & \quad \beta^\top \text{diag}(K(\gamma)) - \beta^\top K(\gamma) \beta \\
n\text{s.t.} & \quad \beta^\top 1 = 1 \quad 0 \leq \beta_i \leq D \quad \forall i
\end{array} \right\} \quad (6)
\]
where $D$ is a regularization parameter. More sophisticated variants can also be adopted by following the idea of support vector data description (SVDD) in [49]. However, these methods will bring one more to-be-tuned hyperparameter $D$ into the trilevel optimization structure. This does not well align with our goal—developing an efficient approach to integrating the radius information into MKL.

III. Proposed $l_2$-norm $\text{tr}(S_T)$ MKL Algorithm

A. Close Relationship Between $R^2$ and $\text{tr}(S_T)$

Recall that $x_i$ ($i = 1, \ldots, n$) denotes the $i$th training sample. The total scatter matrix is defined as $S_T = \sum_{i=1}^{n} (x_i - m)(x_i - m)^\top$, where $m = (1/n) \sum_{i=1}^{n} x_i$ is the sample-based total mean. Although each training sample is implicitly mapped onto a feature space via the kernel trick and $S_T$ in that space is inaccessible, its trace can be explicitly expressed by the kernel function as
\[
\text{tr}(S_T) = \text{tr}(K(\gamma)) - \frac{1}{n} 1^\top K(\gamma) 1 = \sum_{p=1}^{m} \gamma_p a_p \quad (7)
\]
where $a_p = \text{tr}(K_p) - (1/n) 1^\top K_p 1$. The close relationship between $\text{tr}(S_T)$ and the squared radius of the MEB $R^2$ has been revealed in the literature [44]. Both measure the scattering of samples in a kernel-induced feature space, and $\text{tr}(S_T)$ can be shown as an approximation of $R^2$. The detailed analysis on the relationship can be found in [44, Appendix].$^2$ In this paper, instead of incorporating the radius of the MEB directly, we incorporate $\text{tr}(S_T)$, and the advantages are threefold.

1) In the definition of $S_T$, each sample is assigned with an equal weight when measuring data scattering. This makes $\text{tr}(S_T)$ less sensitive to an outlier that significantly deviates from the center of data cloud. In contrast, such an outlier will become an important support vector

$^1$As in SVMs, imposing a box constraint on $\beta$ is equivalent to introducing a slack variable for each training sample in the primal problem of (5).

when optimizing $\beta$ in (5), which makes the obtained $R^2$ sensitive to the outlier and becomes noisy. A similar mechanism has been observed in [35] and used to explain the advantage of kernel Fisher discriminant analysis over SVMs in some situations.

2) Compared to the radius–margin formulation directly adopted in [32], the proposed trace–margin formulation can significantly shorten the kernel learning time by avoiding solving the QP problem required to compute the radius at each iteration.

3) Compared with the radius of the MEB, the trace variant is a linear function of the base-kernel weights $\gamma$, which allows us to prove Theorem 1 in this paper. Based on this theorem, the proposed trace–margin formulation can be effectively transformed to the existing SVM-based MKL formulation, with the only difference being that a weighted norm constraint is used.

Based on the earlier discussion, we substitute $R^2$ with $\text{tr}(S_T)$ in the following to incorporate the radius information into the SVM-based MKL formulation.

### B. Incorporating $\text{tr}(S_T)$ Into MKL

The generalization error of SVMs with a hard margin can be estimated by leave-one-out (LOO) error [50]. This error is upper bounded by the ratio of the radius of the MEB to the margin, called the radius–margin bound in the literature [10], [20], [51]. In order to strictly comply with the radius–margin bound, the $\ell_2$ SVMs with a soft margin have to be used in the presence of nonseparable data. It can be reformulated as the SVMs with a hard margin that employ a slightly modified kernel, making the radius–margin bound still applicable. To better justify the incorporation of radius information, this paper focuses on the $\ell_2$ SVMs with a soft margin.

For a given kernel mapping $\phi(\cdot)$ (or, equivalently, the kernel matrix $K$), the objective of $\ell_2$ SVMs with a soft margin is

$$
\min_{\omega, b, \xi} \frac{1}{2} \left\| \omega \right\|^2 + C \sum_{i=1}^{m} \xi_i
$$

s.t. $y_i \left( \omega^T \phi(x_i) + b \right) \geq 1 - \xi_i, \quad \xi_i \geq 0 \forall i. \quad (8)$

It can be rewritten as the form of SVMs with a hard margin

$$
\min_{\omega, b, \xi} \frac{1}{2} \left\| \omega \right\|^2
$$

s.t. $y_i \left( \omega^T \phi(x_i) + b \right) \geq 1 \forall i \quad (9)$

where $\omega := [\omega, \sqrt{C} \xi]$, $\phi(x_i) := [\phi(x_i), e_i/\sqrt{C}]$, and $e_i$ is a column vector with the $i$th element being one and others being zero. Accordingly, the kernel matrix $K$ will be slightly modified as $K := K + I/C$ due to the modification of the kernel mapping.

To incorporate the radius information into SVM-based MKL, we could straightforwardly minimize the radius–margin bound here to optimize the base-kernel weights. That is

$$
\min_{\gamma, \omega, b} \frac{1}{2} R^2 \left\| \omega \right\|^2
$$

s.t. $y_i \left( \omega^T \phi(x_i; \gamma) + b \right) \geq 1 \forall I \quad \gamma_p \geq 0 \forall p \quad (10)$

where $\phi(x_i; \gamma) := [\phi(x_i; \gamma), e_i/\sqrt{C}]$ and $R^2$ is defined in (5). However, directly optimizing (10) will incur extra computational cost to compute the radius at each iteration. More importantly, the notorious sensitivity of the radius to the outliers in training data will possibly adversely affect its performance in predicting the generalization error.

Following the idea proposed in this paper, $R^2$ is replaced with $\text{tr}(S_T)$, and this leads to the objective function

$$
\min_{\gamma, \omega, b} \frac{1}{2} \text{tr}(S_T) \left\| \omega \right\|^2
$$

s.t. $y_i \left( \omega^T \phi(x_i; \gamma) + b \right) \geq 1 \forall i \quad \gamma_p \geq 0 \forall p \quad (11)$

where $\text{tr}(S_T) = \sum_{p=1}^{m+1} \alpha_p \gamma_p$ according to the definition in (7). Note that, to accommodate the regularization parameter $C$ in (8), an additional base kernel $K_{m+1} = I$ is defined, where $I$ stands for an identity matrix. The kernel weight of $K_{m+1}$ is $\gamma_{m+1} = 1/C$.

It is not difficult to see that (11) can be rewritten as

$$
\min_{\gamma} J(\gamma) \quad \text{s.t.} \quad \gamma_p \geq 0 \forall p \quad (12)
$$

where

$$
J(\gamma) = \left\{ \min_{\omega, b} \frac{1}{2} \text{tr}(S_T) \left\| \omega \right\|^2 \right. $$

s.t. $y_i \left( \omega^T \phi(x_i; \gamma) + b \right) \geq 1 \forall i \} \quad (13)$

This is the optimization problem to be solved in this paper. As aforementioned, one of the goals to incorporating the radius information is to address the scaling issue. In existing SVM-based MKL literature, a remedy to this issue is imposing a norm constraint on the weights of base kernels [14], [33], [43], [47], [48]. However, which kind of norm constraint shall be employed for a given learning task still remains unclear. Note that, in our objective function (12), no norm constraint needs to be imposed on the weights of base kernels except the nonnegativity constraint used to guarantee the combined kernel to be positive semidefinite. Hence, the proposed approach maintains the merit of incorporating the radius information achieved by the work [32].

To formally show this, the following three propositions are proved. Recall that $\kappa = \{k_1, \ldots, k_m, k_m+1\}$ denotes a group of base kernels and $k_{m+1}(x_i, x_j) = \delta(i - j)$, where $\delta(0) = 1$ and 0 otherwise. Let $d_1, \ldots, d_m, d_{m+1}$ be any positive scalars. We have the following propositions.

**Proposition 1:** $J(\tau \gamma) = J(\gamma)$, where $\tau$ is any positive scalar, and the SVM decision function is not affected by $\tau$.

**Proof:** By the definition of $\phi(x_i; \gamma)$, we have

$$
\phi(x_i; \gamma) = [\sqrt{\gamma_1} \phi_1(x_i), \ldots, \sqrt{\gamma_{m+1}} \phi_{m+1}(x_i)]^T. \quad (14)
$$

Hence, it can be verified that

$$
\phi(x_i; \tau \gamma) = \sqrt{\tau} \phi(x_i; \gamma). \quad (15)
$$
From (7), (15), and (13), we have
\[
\mathcal{J}(\gamma) = \left\{ \min_{\omega_b} \frac{1}{2} \text{tr}(S_T) \| \omega \|^2 \right\},
\]
\[
\text{s.t. } y_i \left( \sqrt{\tau} \omega^T \phi(x_i; \gamma) + b \right) \geq 1 \forall i.
\]
(16)
By defining \( \bar{\omega} = \sqrt{\tau} \omega \), (16) can be rewritten as
\[
\mathcal{J}(\gamma) = \left\{ \min_{\bar{\omega}_b} \frac{1}{2} \text{tr}(S_T) \| \bar{\omega} \|^2 \right\},
\]
\[
\text{s.t. } y_i \left( \bar{\omega}^T \phi(x_i; \gamma) + b \right) \geq 1 \forall i
\]
\[
= \mathcal{J}(\gamma)
\]
(17)
which indicates that \((\omega^* / \sqrt{\tau}, b^*)\) will be the optimal solution of
(16) if \((\omega^*, b^*)\) is the optimal solution of (13). It is not difficult to verify that the decision function of SVMs will be
\( f(x) = \omega^T \phi(x; \gamma) + b^* \), which will not vary with \( \tau \). This completes the proof. ■

By Proposition 1, we can observe that our objective in (12) is invariant when the weights of base kernels are uniformly scaled by a positive scalar \( \tau \). This is because, in such a case, the optimal value of \( \omega \) will also be scaled by \( 1 / \tau \) accordingly. The two changes cancel each other and keep the SVM decision function as it was. Based on this property, we further show that our formulation in (12) is invariant with respect to different types of norm constraints in Proposition 2.

**Proposition 2:** Let the problem (P1) be
\[
\min_{\gamma} \mathcal{J}(\gamma) \text{ s.t. } \gamma_p \geq 0 \forall p
\]
and the problem (P2) be
\[
\min_{\gamma} \mathcal{J}(\gamma) \text{ s.t. } \gamma_p \geq 0 \forall p \quad \| \gamma \|_q = 1
\]
where \( \| \cdot \|_q \) stands for a norm. The following can be concluded:
1) Denoting an optimal solution of (P1) as \( \gamma^* \), \( \gamma^*_1 / \| \gamma^*_1 \|_q \)
must be the optimal solution of (P2), and 2) the optimal solution of (P2), denoted as \( \gamma^*_2 \), must be the optimal solution of (P1).

**Proof:** Because \( \gamma^*_1 \) is an optimal solution of (P1), there must be \( \delta > 0 \) such that, for any \( \gamma(\gamma_p \geq 0 \forall p) \) satisfying \( \| \gamma - \gamma^*_1 \|_q \leq \delta \), it can be obtained that \( \| \gamma - \gamma^*_1 \|_q \leq \delta \), and this leads to \( \mathcal{J}(\gamma) \leq \mathcal{J}(\gamma^*_1) \). According to Proposition 1, both \( \mathcal{J}(\gamma^*_1) = \mathcal{J}(\gamma^*_1 / \| \gamma^*_1 \|_q) \) and \( \mathcal{J}(\gamma^*_1 / \| \gamma^*_1 \|_q) = \mathcal{J}(\gamma) \) are true. Hence, it can be obtained that, for any \( \gamma \) that satisfies \( \| \gamma - (\gamma^*_1 / \| \gamma^*_1 \|_q) \|_2 \leq \delta \), there must be \( \| \gamma - (\gamma^*_1 / \| \gamma^*_1 \|_q) \|_q \leq \delta \), which implies \( \| \gamma - (\gamma^*_1 / \| \gamma^*_1 \|_q) \|_q \leq \delta \). Noting that \( \gamma^*_1 / \| \gamma^*_1 \|_q \) satisfies the constraint of (P2), it can be concluded that \( \gamma^*_1 / \| \gamma^*_1 \|_q \) is an optimal solution of (P2). This proves the conclusion (1).

Because \( \gamma^*_2 \) is an optimal solution of (P2), there must be \( \delta > 0 \) such that, for any \( \gamma(\gamma_p \geq 0 \forall p, \| \gamma \|_q = 1) \) satisfying \( \| \gamma - \gamma^*_2 \|_2 \leq \delta \), \( \mathcal{J}(\gamma^*_2) \leq \mathcal{J}(\gamma) \) is true. Note that, for any \( \gamma(\gamma_p \geq 0 \forall p) \) in the feasible domain of (P1), \( \gamma / \| \gamma \|_q \) must be in the feasible domain of (P2). Therefore, for any \( \gamma \) satisfying \( \| \gamma / \| \gamma \|_q \|_q \leq \delta \), it can be obtained that \( \mathcal{J}(\gamma^*_2) \leq \mathcal{J}(\gamma / \| \gamma \|_q) \). Noting that \( \gamma^*_2 \) satisfies the constraint of (P1), it can be concluded that \( \gamma^*_2 \) is an optimal solution of (P1). This proves the conclusion (2). ■

Built upon the invariance with respect to norm constraints shown in Proposition 2, the following further proves that our formulation in (12) is invariant to any scaling applied to the base-kernel weights at the initial optimization.

**Proposition 3:** Let the problem (P3) be
\[
\min_{\gamma} \mathcal{J}(\gamma) \text{ s.t. } \gamma_p \geq 0 \forall p \quad \| \gamma \|_q = 1
\]
and the problem (P4) be
\[
\min_{\gamma} \mathcal{J}(\gamma \otimes d) \text{ s.t. } \gamma_p \geq 0 \forall p \quad \| \gamma \|_q = 1
\]
where \( \| \cdot \|_q \) stands for a norm. The following can be concluded: 1) (P3) and (P4) have the same optimal function value, and 2) denoting an optimal solution of (P4) as \( \gamma^*_2 \), \( \| \gamma^*_2 \|_q \) must be the optimal solution of (P3), where \( d = (d_1, \ldots, d_{m+1})^{-1} \) and \( \otimes \) denotes componentwise product.

**Proof:** By removing the norm constraint in (P4), a new problem (P5) is defined as \( \min_{\gamma} \mathcal{J}(\gamma \otimes d) \) s.t. \( \gamma_p \geq 0 \forall p \). According to Propositions 1 and 2, (P4) and (P5) shall have the same optimal function value. By defining \( \tilde{\gamma} = \gamma \otimes d \), problem (P5) is rewritten as (P6) \( \min_{\tilde{\gamma}} \mathcal{J}(\tilde{\gamma}) \), s.t. \( \tilde{\gamma}_p \geq 0 \forall p \). Note that the solutions of (P5) and (P6) are bijective because the linear transform between \( \gamma \) and \( \tilde{\gamma} \) is invertible. Applying Propositions 1 and 2 again, it can be obtained that imposing a norm constraint onto problem (P6) does not change its optimal function value. Since (P6) with a norm constraint is equivalent to problem (P3), it can be concluded that (P3) and (P4) have the same function value. This proves the conclusion (1).

Applying Proposition 2, it is obtained that if \( \gamma^*_2 \) is an optimal solution of (P4), then \( \gamma^*_2 \) must be the optimal solution of (P5). Due to the bijection between the solutions of (P5) and (P6), \( \gamma^* \otimes d \) must be the optimal solution of (P6). Again, by Proposition 2, it can be concluded that \( \gamma^* \otimes d \) must be the optimal solution of (P3). This proves the conclusion (2). ■

Proposition 3 implies that, with any initial scaling of the weights of base kernels, our formulation in (12) will end up with the same kernel combinations up to the scaling factor \( \| \gamma^*_2 \|_q \). Again, by Proposition 1, we know that this will keep the SVM decision function unchanged. Through the aforementioned three propositions, we show that our formulation in (12) will achieve the same optimal decision function no matter how the weights of base kernels are scaled. This indicates that our formulation can automatically handle the scaling and initialization problem pointed out in [32].

Now, we show that our objective in (12) can be reformulated into the form similar to the existing SVM-based MKL algorithms, such as SimpleMKL [18] and Shogun [45], and thus be readily solved. The only difference is that a weighted norm constraint on the weights of base kernels is used in our algorithm, while an unweighted norm constraint is used in [18] and [45].
Theorem 1: Let $\gamma^*$ denote the optimal solution of the optimization problem in (12). It can be written as $\gamma^* = \text{tr}(S_T)\eta^*$, where $\eta^*$ is the optimal solution of the following optimization problem:

$$\min_{\eta} J(\eta)$$

s.t. $\sum_{p=1}^{m+1} a_p \eta_p = 1$, $\eta_p \geq 0 \forall p$. \hfill (18)

where

$$J(\eta) = \left\{ \min_{\omega, b} \frac{1}{2} \|\omega\|^2 \right\}$$

s.t. $y_i \left( \omega^\top \phi(x_i; \eta) + b \right) \geq 1 \forall i$. \hfill (19)

Proof: By defining $\tilde{\omega}^\Delta = \sqrt{\text{tr}(S_T)} \omega$, (13) can be transformed to

$$J(\gamma) = \left\{ \min_{\omega, b} \frac{1}{2} \|\tilde{\omega}\|^2 \right\}$$

s.t. $y_i \left( \tilde{\omega}^\top \phi \left( x_i; \frac{\gamma}{\text{tr}(S_T)} \right) + b \right) \geq 1 \forall i$. \hfill (20)

Then, defining $\eta^\Delta = \gamma / \text{tr}(S_T)$ and noting that $\text{tr}(S_T) = \sum_{p=1}^{m+1} a_p \gamma_p$, it can be obtained that $\sum_{p=1}^{m+1} a_p \eta_p = 1$. Also, according to Proposition 1, $J(\gamma) = J(\gamma / \text{tr}(S_T)) = J(\eta)$ is true. As a result, (20) can be written as

$$J(\eta) = \left\{ \min_{\omega, b} \frac{1}{2} \|\tilde{\omega}\|^2 \right\}$$

s.t. $y_i \left( \tilde{\omega}^\top \phi \left( x_i; \eta \right) + b \right) \geq 1 \forall i$

$$\sum_{p=1}^{m+1} a_p \eta_p = 1$, $\eta_p \geq 0 \forall p$. \hfill (21)

This is essentially the optimization problem in (18). Solving this problem and denoting the optimal solution as $\eta^*$, the optimal solution of (12) can be obtained as $\gamma^* = \text{tr}(S_T)\eta^*$. This completes the proof.

The aforementioned theorem indicates that our formulation in (12) can be solved by the problem in (18). Note that the latter shares a similar form with SimpleMKL [18] except the weighted $\ell_1$-norm constraint. Through such an equivalence, the formulation in (12) can thus be conveniently solved with the method used in [18], as shown in the next section.

C. Algorithm

The dual problem of (19) is

$$J(\eta) = \left\{ \max_{\alpha} 1^\top \alpha - \frac{1}{2} (\alpha \otimes y)^\top \left( \sum_{p=1}^{m+1} \eta_p K_p \right) (\alpha \otimes y) \right\}$$

s.t. $\alpha^\top y = 0$, $\alpha_i \geq 0 \forall i$. \hfill (22)

As aforementioned, the optimization problem (18) can be solved by any existing SVM-based MKL packages. This paper follows SimpleMKL [18] to use the reduced gradient method. The weights of base kernels $\eta$ and the structural parameters of SVMs $\alpha$ are updated alternately until convergence. The algorithm is outlined in Algorithm 1.

Algorithm 1: tr(S_T) MKL ($\ell_2$trStMKL)

1: $\eta_0 = (1/(m+1))$, $a_p = \text{tr}(K_p) - (1/n)1^\top K_p 1$, $p = 1, \ldots, m+1$, $K_{m+1} = I$ and $\gamma_{m+1} = 1/C$.
2: $i \leftarrow 0$
3: repeat
4: Obtain $\alpha^{i+1}$ by solving (22) with $\eta^i$.
5: Update $\eta^{i+1}$ based on $\alpha^{i+1}$ by reduced gradient method.
6: $i \leftarrow i + 1$
7: until Convergence

Recalling Proposition 1, we know that problems in (12) and (18) share the same SVM decision function. Therefore, after we obtain the optimal $\alpha^*$, $b^*$, and $\eta^*$ by solving (18), we can write the SVM decision function as

$$f(x) = \sum_{i=1}^{n} \alpha^*_i y_i \left( \sum_{p=1}^{m} \eta^*_p k_p(x, x) \right) + b^*$$. \hfill (23)

D. Discussion

It is shown in [44] that $\text{tr}(S_T)/n$ is a lower bound of $R^2$. As a result, our trace–margin criterion, i.e., $\text{tr}(S_T)/\|\omega\|^2$, may not necessarily be an upper bound of LOO error like the radius–margin bound, i.e., $R^2/\|\omega\|^2$. However, we observe that the trace–margin criterion may correlate well with the generalization performance. In order to investigate this point, we compare the two criteria, radius–margin bound and trace–margin criterion, with respect to the generalization error on a controlled toy data set sampled from two Gaussian distributions of $N(\mu_1, I)$ and $N(\mu_2, 2I)$, where $\mu_1 = [0, 1]^T$, $\mu_2 = [2, 0]^T$, and $I$ is a $2$-D identity matrix. We randomly sample 20 groups of training data and test data. Each training data set and test data set contains 100 and 1000 samples, respectively. Gaussian and polynomial kernels are used as the base kernels. Then, we compute $R^2/\|\omega\|^2$, $\text{tr}(S_T)/\|\omega\|^2$, and the generalization error by applying a different kernel combination coefficient. The regularization parameter $C$ that minimizes the generalization error is equally applied to compute $\text{tr}(S_T)/\|\omega\|^2$ and $R^2/\|\omega\|^2$. The averaged results on 20 groups of data sets are reported. Also, the correlation coefficient between $R^2/\|\omega\|^2$ and the generalization error and that between $\text{tr}(S_T)/\|\omega\|^2$ and the generalization error are computed. Fig. 1 plots the curve of $R^2/\|\omega\|^2$, $\text{tr}(S_T)/\|\omega\|^2$, and the generalization error with respect to the base-kernel combination coefficient.

Fig. 1(a) shows the case of combining two Gaussian kernels having different widths, one-half and two, respectively. The correlation coefficient between $\text{tr}(S_T)/\|\omega\|^2$ and the generalization
error is 0.7537, and the correlation coefficient between the generalization error and the base-kernel combination coefficient. In (a), two Gaussian kernels, with widths of one-half and two, respectively, are used as base kernels. The correlation coefficient between \( \| S_T \| \omega \|^2 \) and the generalization error is 0.7537, and the correlation coefficient between \( R^2 \| \omega \|^2 \) and the generalization error is 0.7446. In (b), a polynomial kernel with a degree of two and a Gaussian kernel with a width of one-half are used as base kernels. The correlation coefficient between \( \| S_T \| \omega \|^2 \) and the generalization error is 0.4919, while the correlation coefficient between \( R^2 \| \omega \|^2 \) and the generalization error becomes 0.1413. The same \( C \) that minimizes the generalization error is equally applied to compute the two criteria. (a) Gaussian + Gaussian. (b) Poly + Gaussian.

error is 0.7537, and the correlation coefficient between \( R^2 \| \omega \|^2 \) and the generalization error is 0.7446. As seen, the trace–margin criterion correlates well with the generalization error as the radius–margin bound does. Fig. 1(b) shows the case of combining a Gaussian kernel with the width of one-half and a polynomial kernel with a degree of two. The correlation coefficient between \( \| S_T \| \omega \|^2 \) and the generalization error is 0.4919, while the value for \( R^2 \| \omega \|^2 \) is 0.1413. From Fig. 1, we can observe that \( \| S_T \| \omega \|^2 \) correlates well with the generalization error and performs better than \( R^2 \| \omega \|^2 \) in the second case. This example preliminarily demonstrates the good property of our trace–margin criterion. Its superior performance will be further verified through extensive experimental study in Section IV.

### IV. EXPERIMENTAL RESULTS

In this section, we compare our proposed \( \ell_2 \)trStMKL with state-of-the-art MKL algorithms including SimpleMKL [18], the algorithm in [32] [Minimum Ball Multiple Kernel Learning (MBMKL)], the algorithm in [31] [Radius Multiple Kernel Learning (RMKL)], the discriminant MKL (DMKL) [36], nonsparse MKL [14] (\( \ell_p \)-MKL with \( p = 4/3, 2, \) and 4, respectively), uniform-weight MKL (UWMKL), and the best single-kernel SVMs with cross-validation. We implement our \( \ell_2 \)trStMKL algorithm, MBMKL, RMKL, \( \ell_p \)-MKL, and UWMKL based on the SimpleMKL package\(^4\) by following the corresponding papers. The codes for DMLK are downloaded from the authors’ web site.\(^4\) The aforementioned MKL algorithms are tested on 11 UCI data sets used in [32], three protein subcellular localization data sets [52], and the Caltech101 data set [53]. We will verify that the proposed \( \ell_2 \)trStMKL is able to achieve the overall best classification performance, particularly in the presence of outliers. Also, in addition to avoiding time-consuming cross-validation procedure, our algorithm maintains the computational efficiency of the existing SVM-based MKL formulation and requires less computation than the radius-incorporated methods in [31] and [32]. The experiments are conducted on a high-performance cluster server, where each node has a 2.3-GHz CPU and 16-GB memory.

#### A. Experimental Results on UCI Data Sets

We compare the aforementioned ten algorithms on the 11 UCI data sets used in [32]. All the experiments use the following setting: For each data set, 40% of the data is used for training and the rest is used for test. The regularization parameter \( C \) in SimpleMKL, MBMKL, RMKL, \( \ell_p \)-MKL (\( p = 4/3, 2, \) and 4), UWMKL, and the best single-kernel SVMs is chosen from a sufficiently large range \( \{2^{-5}, 2^{-3}, \ldots, 2^{15}\} \), according to [54] by fourfold cross-validation on each training set. For our proposed \( \ell_2 \)trStMKL, the regularization parameter \( C \) is treated as an extra kernel parameter and tuned automatically. All data sets have been normalized to have zero mean and unit variance on each feature by following [17]–[19]. The base kernels are the same as those in [18] and [32], which include ten Gaussian kernels with the widths of \( [0.5, 1, 2, 5, 7, 10, 12, 15, 17, 20] \) and ten polynomial kernels with degrees of one to ten. Each experiment is repeated 30 times, and the average accuracy and standard deviation are reported.

To conduct a rigorous comparison, the paired Student’s \( t \)-test\(^3\) is performed. The \( p \)-value of the \( t \)-test represents the probability that two sets of compared results come from the distributions with an equal mean. A \( p \)-value of 0.05 is considered statistically significant. The results are listed in Table I, where the highest accuracy and those whose differences from the highest accuracy are not statistically significant are shown in bold for each data set. The corresponding time cost (including both cross-validation time and training time) is also listed in Table II.

---

\(^3\)http://asi.insa-rouen.fr/enseignants/arakotom/code/mklindex.html.

\(^4\)http://www.public.asu.edu/jye02/Software/index.html.
We also design an experiment to compare the robustness of the aforementioned radius-incorporated algorithms in the presence of outliers. To generate an outlier, we find the sample whose Euclidean distance from the mean of all samples is the largest one for each training set. With this sample, we create a set of outliers by multiplying its feature vector with a scalar of \{5, 10, 15\}, respectively. An outlier is then added to a training set, and the aforementioned algorithms are compared on it. All the other experimental settings are the same as the aforementioned settings. The corresponding classification accuracy and the statistical test result are presented in Tables III–V, respectively.

From these experimental results, we have the following observations.

1) Classification accuracy. In Tables I and III–V, the proposed \(\ell_2\)trStMKL consistently achieves the overall best classification performance. In the standard setting where no outlier is added, the performance of our \(\ell_2\)trStMKL is among the best on seven out of the whole 11 UCI data sets, which is verified by the \(p\)-value of the pairwise \(t\)-test shown in Table I. In the presence of an outlier, our \(\ell_2\)trStMKL shows clear improvement on the other algorithms, and its classification performance is among the best on seven, eight, and nine of the whole 11 data sets, respectively, as shown in Tables III–V. The main difference of these algorithms in comparison lies at that our \(\ell_2\)trStMKL uses \(\text{tr}(S^T)\) to characterize the radius information, while \(\text{tr}(K)\) is used in SimpleMKL \[18\] and \(\ell_p\)-MKL, and the radius of the MEB is used in MBMKL \[32\] and RMKL \[31\]. The aforementioned experimental result indicates the efficiency of employing \(\text{tr}(S^T)\) in our \(\ell_2\)trStMKL to incorporate the radius information.

2) Robustness. Although all the algorithms in comparison show degraded classification accuracy with the increasing magnitude of the outlier (see Tables III–V), our \(\ell_2\)trStMKL still consistently gives the best classification performance. This result demonstrates the robustness of \(\text{tr}(S_T)\) to outliers. By assigning each point the same weight, \(\text{tr}(S_T)\) computes an average squared radius of data scattering, which is more stable in the presence of outliers. In contrast, when computing the radius of the MEB, a point farther away from the center of data cloud would have a greater weight.
often assigned larger weight, making the radius sensitive to outliers.

3) Computational efficiency. Table II demonstrates the computational efficiency of our \( \ell_2 \)-trStMKL. The timing result listed in Table II includes cross-validation time and training time. The average time costs by SimpleMKL, MBMKL, RMKL, \( \ell_p \)-MKL (\( p = 4/3, 2, \) and 4), UWMKL, and the best single-kernel SVMs are about 19.2, 88.2, 699.0, 96.1, 45.0, 24.1, 1.7, and 38.8 times longer than those used by \( \ell_2 \)-trStMKL. Comparing with SimpleMKL, MBMKL, RMKL, \( \ell_p \)-MKL (\( p = 4/3, 2, \) and 4), UWMKL, and the best single-kernel SVMs, \( \ell_2 \)-trStMKL does not need cross-validation to tune the regularization parameter \( C \). Instead, it automatically learns the optimal \( C \) value by treating it as an extra kernel weight. Also, comparing with MBMKL, our \( \ell_2 \)-trStMKL avoids the extra QP needed to compute the radius of the MEB at each iteration. The two factors make our proposed \( \ell_2 \)-trStMKL more computationally efficient than the other algorithms.

In sum, the aforementioned experimental results indicate that our \( \ell_2 \)-trStMKL achieves the overall best classification performance among the algorithms in comparison with the least computational overhead. In terms of the ratio of classification performance to time cost, the proposed \( \ell_2 \)-trStMKL is clearly the best one.
TABLE V

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>colo.</td>
<td>81.2 ± 4.9 (1.00)</td>
<td>67.8 ± 6.1 (0.00)</td>
<td>77.5 ± 6.3 (0.00)</td>
<td>70.8 ± 6.9</td>
<td>71.9 ± 8.0 (0.00)</td>
<td>69.1 ± 4.6 (0.00)</td>
<td>66.0 ± 2.2 (0.00)</td>
<td>71.0 ± 4.9 (0.00)</td>
</tr>
<tr>
<td>four.</td>
<td>97.3 ± 1.6 (0.00)</td>
<td>98.0 ± 1.7 (0.00)</td>
<td>99.4 ± 0.7 (1.00)</td>
<td>98.6 ± 1.9 (0.00)</td>
<td>92.9 ± 5.9 (0.00)</td>
<td>98.0 ± 1.6 (0.00)</td>
<td>98.1 ± 1.7 (0.00)</td>
<td>98.1 ± 1.6 (0.00)</td>
</tr>
<tr>
<td>germ.</td>
<td>73.4 ± 1.1 (0.07)</td>
<td>72.0 ± 1.3 (0.00)</td>
<td>71.5 ± 1.0 (0.00)</td>
<td>71.7 ± 1.9 (0.00)</td>
<td>73.7 ± 1.5 (0.10)</td>
<td>73.6 ± 1.2 (0.02)</td>
<td>74.0 ± 1.3 (1.00)</td>
<td>73.9 ± 1.3 (0.42)</td>
</tr>
<tr>
<td>iono.</td>
<td>94.3 ± 1.2 (0.12)</td>
<td>94.2 ± 1.2 (0.01)</td>
<td>94.1 ± 1.2 (0.01)</td>
<td>93.6 ± 1.3 (0.00)</td>
<td>94.0 ± 1.0 (0.00)</td>
<td>94.3 ± 1.0 (0.06)</td>
<td>94.6 ± 1.0 (1.00)</td>
<td>94.5 ± 1.0 (0.29)</td>
</tr>
<tr>
<td>liver.</td>
<td>61.3 ± 3.2 (0.00)</td>
<td>64.7 ± 4.0 (0.00)</td>
<td>66.2 ± 3.8 (0.00)</td>
<td>64.7 ± 4.7 (0.00)</td>
<td>64.2 ± 4.1 (0.00)</td>
<td>63.9 ± 3.7 (0.00)</td>
<td>64.2 ± 4.1 (0.00)</td>
<td>64.2 ± 3.7 (0.00)</td>
</tr>
<tr>
<td>somn.</td>
<td>82.0 ± 3.2 (0.14)</td>
<td>82.1 ± 3.3 (0.00)</td>
<td>82.5 ± 3.5 (1.00)</td>
<td>78.8 ± 6.1 (0.00)</td>
<td>82.5 ± 3.1 (0.45)</td>
<td>81.5 ± 3.5 (0.07)</td>
<td>80.3 ± 4.0 (0.00)</td>
<td>80.5 ± 3.7 (0.01)</td>
</tr>
<tr>
<td>spli.</td>
<td>83.6 ± 1.1 (0.87)</td>
<td>83.0 ± 1.6 (0.00)</td>
<td>83.7 ± 1.5 (1.00)</td>
<td>80.9 ± 2.6 (0.00)</td>
<td>83.0 ± 1.5 (0.49)</td>
<td>82.7 ± 1.6 (0.00)</td>
<td>82.8 ± 1.2 (0.00)</td>
<td>82.8 ± 1.1 (0.00)</td>
</tr>
<tr>
<td>wdbc.</td>
<td>96.6 ± 0.7 (0.42)</td>
<td>95.5 ± 1.1 (0.00)</td>
<td>95.6 ± 1.1 (0.00)</td>
<td>95.0 ± 1.4 (0.00)</td>
<td>95.8 ± 0.8 (0.00)</td>
<td>95.7 ± 1.0 (0.00)</td>
<td>96.0 ± 0.9 (0.05)</td>
<td>96.2 ± 0.8 (0.18)</td>
</tr>
<tr>
<td>wpbc.</td>
<td>77.7 ± 1.9 (0.50)</td>
<td>76.3 ± 1.4 (0.01)</td>
<td>76.0 ± 1.8 (0.00)</td>
<td>76.2 ± 1.2 (0.00)</td>
<td>76.1 ± 2.3 (0.00)</td>
<td>76.7 ± 2.9 (0.11)</td>
<td>77.4 ± 2.9 (1.00)</td>
<td>77.9 ± 3.2 (0.00)</td>
</tr>
<tr>
<td>hear.</td>
<td>81.7 ± 1.8 (0.07)</td>
<td>78.5 ± 2.4 (0.00)</td>
<td>79.5 ± 2.8 (0.00)</td>
<td>80.0 ± 2.4 (0.00)</td>
<td>80.4 ± 2.5 (0.00)</td>
<td>78.9 ± 2.9 (0.00)</td>
<td>79.3 ± 2.3 (0.00)</td>
<td>79.7 ± 2.4 (0.00)</td>
</tr>
<tr>
<td>musk.</td>
<td>84.1 ± 2.7 (0.08)</td>
<td>82.2 ± 2.6 (0.00)</td>
<td>82.8 ± 2.1 (0.00)</td>
<td>78.2 ± 6.9 (0.00)</td>
<td>83.7 ± 2.6 (0.00)</td>
<td>82.8 ± 3.1 (0.00)</td>
<td>83.3 ± 2.8 (0.00)</td>
<td>84.2 ± 2.6 (1.00)</td>
</tr>
<tr>
<td>avg.</td>
<td>85.9</td>
<td>81.3</td>
<td>82.6</td>
<td>80.8</td>
<td>81.7</td>
<td>81.6</td>
<td>81.5</td>
<td>82.1</td>
</tr>
<tr>
<td>win.</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 2. Classification accuracy comparison.

B. Experimental Results on Protein Subcellular Localization Data Sets

Protein subcellular localization plays a pivotal role in the tasks of protein function prediction and protein interactions. Three protein subcellular localization data sets including plant, PsortPos, and PsortNeg have been widely used as benchmark data sets to test the performance of MKL algorithms [33], [52]. The class numbers of these three data sets are 4, 4, and 5, respectively. For these protein subcellular localization data sets, we follow the setup of [33] and [52] by the following: 1) constructing 69 kernels: 5 two kernels on phylogenetic trees, three kernels from BLAST E-values, and 64 sequence motif kernels; 2) normalizing each base kernel; and 3) measuring the performance of each MKL algorithm by classification accuracy, Matthews correlation coefficient (MCC), and $F_1$ score. In this experiment, we omit the nonsparse MKL algorithms because it has been reported in [14] that sparse MKL algorithms can usually achieve a better performance.

5Available at http://mkl.ucsd.edu/dataset/protein-subcellular-localization.

Fig. 3. MCC comparison.

Fig. 4. $F_1$ score comparison.
Following [33], we randomly split each of the protein subcellular localization data sets into two parts equally, where half of the data is used for training and the rest is for test. This procedure is repeated by ten times, and the average results are reported. We use the one-versus-rest strategy to handle multiclass classification problems. Figs. 2–4 report the average results on ten runs in terms of classification accuracy, MCC, and $F_1$ score, respectively. Our $\ell_2\text{trStMKL}$ achieves the best results on all three data sets. Take the plant data set for example. The classification accuracy obtained by our $\ell_2\text{trStMKL}$ is 91.7%, compared to 90.4% obtained by SimpleMKL, 90.7% by MBMKL, 90.8% by RMKL, and 88.4% by UWMKL. The MCC obtained by our $\ell_2\text{trStMKL}$ is 85.4%, compared to 84.2% by SimpleMKL, 83.9% by MBMKL, 83.5% by RMKL, and 79.4% by UWMKL. The $F_1$ score obtained by our $\ell_2\text{trStMKL}$ is 88.2%, compared to 87.3% obtained by SimpleMKL, 87.0% by MBMKL, 86.9% by RMKL, and 84.0% by UWMKL. Similar results can also be observed from the other two data sets. The results indicate that our proposed $\ell_2\text{trStMKL}$ improves the classification performance consistently and significantly compared to other radius-incorporated MKL algorithms on all three protein data sets. For the computational efficiency, by avoiding cross-validation and the extra QP, our proposed $\ell_2\text{trStMKL}$ saves much training time.

C. Experimental Results on Caltech-101

The Caltech-101-MKL data set\(^6\) is a collection of kernels derived from various visual features computed on the Caltech-101 object recognition task by using 15 training and 15 test examples per object class. It is a benchmark data set for MKL with the aforementioned features for five random splits of training and test sets. Table VI reports the result of each split and the average result. In addition to the aforementioned MKL algorithms, the optimal results obtained by a single base kernel are also included from the aforementioned web site. Our $\ell_2\text{trStMKL}$ achieves the best average classification accuracy, 68.6%, compared to 63.7% obtained by SimpleMKL, 68.3% by MBMKL, 64.8% by RMKL, 56.0% by DMKL, 65.0% by $\ell_4/3$-MKL, 65.3% by $\ell_2$-MKL, 65.1% by $\ell_4$-MKL, 65.0% by UWMKL, and 60.1% by the optimal single base kernel. Our $\ell_2\text{trStMKL}$ shows a clear improvement on other compared algorithms.

In addition to showing a slightly better classification accuracy than MBMKL, our $\ell_2\text{trStMKL}$ is computationally much more efficient than MBMKL by avoiding the extra QP and the time-consuming cross-validation procedure. These two factors, particularly the latter, make MBMKL less computationally efficient and even prohibit it from large-scale applications. For the Caltech-101-MKL data set, MBMKL spends more than one week on a node of the high-performance cluster server in finishing the cross-validation step to tune the regularization parameter $C$. Moreover, after obtaining the optimal parameter $C$, MBMKL [32] takes another 9953.4 s to learn the kernel combination weights and train the SVM classifier, while our $\ell_2\text{trStMKL}$ only needs 3182.6 s to complete the whole procedure, including both tuning the parameter $C$ and training the SVM classifier. Considering the ratio of classification accuracy to time cost, our $\ell_2\text{trStMKL}$ is clearly the best one, and this puts it in a better position to handle large-scale data sets in real applications. In a broader sense, our $\ell_2\text{trStMKL}$ provides a more efficient approach to incorporating the radius information into the existing MKL formulations.

V. Conclusion

In this paper, an efficient approach to integrating radius information into existing MKL algorithms has been proposed. By taking advantage of the relationship between the trace of data scattering matrix and the radius of the MEB, we have incorporated the radius information by employing the trace of the total scattering matrix of training data. Our theoretical analysis has shown that the resulting optimization in the proposed algorithm can be transformed to a form commonly used in existing MKL literature, and the only difference is that a weighted norm constraint is imposed on the weights of base kernels. Extensive experiments have been conducted on UCI, protein subcellular localization, and Caltech-101 data.

\(^6\)Available at http://mkl.ucsd.edu/dataset/ucsd-mit-caltech-101-mkl-dataset.
sets. As experimentally demonstrated, our algorithm has given the overall best classification performance and the highest computational efficiency among the compared algorithms. Our work provides MKL with an efficient way to utilizing the radius information.

Much future work is worth exploring. For example, the soft-margin MEB (known as SVDD [49]) can be adopted to improve the robustness of the radius of the MEB. As pointed out by an anonymous reviewer, the incorporation of SVDD has the following nice effects.

1) The work in [32] can be recovered as a special case.
2) Our proposed trace–margin approach can also be recovered.
3) A variety of new formulations by varying the softness parameter of the SVDD can be obtained.

We will explore this approach in our future work to further improve the classification performance.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and the associate editor for their constructive comments and guidance that helped to improve the manuscript.

REFERENCES


[8] F. Young and H. Chang, “A kernel approach for semisupervised metri-

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.


Jianping Yin received the M.S. degree and the Ph.D. degree in computer science from the National University of Defense Technology, Changsha, China, in 1986 and 1990, respectively.

He is a Full Professor of computer science with the National University of Defense Technology. His research interests involve artificial intelligence, pattern recognition, algorithm design, and information security.

En Zhu received the M.S. degree and the Ph.D. degree in computer science from the National University of Defense Technology, Changsha, China, in 2001 and 2005, respectively.

From July 2009 to July 2010, he visited the Department of Computer Science, The University of York, York, U.K. He is with the School of Computer, National University of Defense Technology. His main research interests are pattern recognition, image processing, and machine learning.

Xinwang Liu (S’12) received the M.S. degree in computer science from the National University of Defense Technology, Changsha, China, in 2008. He is currently working toward the Ph.D. degree at the National University of Defense Technology and in the School of Computer Science and Software Engineering, University of Wollongong, Wollongong, Australia.

From October 2010, he spent one year in visiting the College of Engineering and Computer Science, Australia National University, Canberra, Australia, supported by the China Scholarship Council. His research interests focus on kernel learning and feature selection.

Lei Wang (SM’10) received the B.Eng. and M.Eng. degrees from Southeast University, Nanjing, China, in 1996 and 1999, respectively, and the Ph.D. degree from Nanyang Technological University, Singapore, Singapore, in 2004.

He is currently a Senior Lecturer with the Faculty of Informatics, University of Wollongong, Wollongong, Australia. His research interests include machine learning, pattern recognition, and computer vision.

Dr. Wang was a recipient of the Australian Postdoctoral Fellowship by the Australian Research Council in 2007 and the Early Career Researcher Award by the Australian Academy of Science in 2009.

Dr. Zhang is an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and EURASIP Journal on Image and Video Processing. He is a General Cochair to host the IEEE International Conference on Multimedia and Expo in Melbourne, Australia, 2012.