Neurocomputing 149 (2015) 253–264

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

# Multiple kernel extreme learning machine

Xinwang Liu [a,*], Lei Wang [b], Guang-Bin Huang [c], Jian Zhang [d], Jianping Yin [a]

[a] School of Computer Science, National University of Defense Technology, Changsha 410073, China
[b] School of Computer Science and Software Engineering, University of Wollongong, NSW 2522, Australia
[c] School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore
[d] Faculty of Engineering and Information Technology, University of Technology, Sydney, NSW 2007, Australia

ARTICLE INFO

*Article history:*
Received 2 August 2013
Received in revised form
18 September 2013
Accepted 20 September 2013
Available online 6 September 2014

*Keywords:*
Extreme learning machine
Multiple kernel learning
Support vector machines

ABSTRACT

Extreme learning machine (ELM) has been an important research topic over the last decade due to its high efficiency, easy-implementation, unification of classification and regression, and unification of binary and multi-class learning tasks. Though integrating these advantages, existing ELM algorithms pay little attention to optimizing the choice of kernels, which is indeed crucial to the performance of ELM in applications. More importantly, there is the lack of a general framework for ELM to integrate multiple heterogeneous data sources for classification. In this paper, we propose a general learning framework, termed multiple kernel extreme learning machines (MK-ELM), to address the above two issues. In the proposed MK-ELM, the optimal kernel combination weights and the structural parameters of ELM are jointly optimized. Following recent research on support vector machine (SVM) based MKL algorithms, we first design a sparse MK-ELM algorithm by imposing an $\ell_1$-norm constraint on the kernel combination weights, and then extend it to a non-sparse scenario by substituting the $\ell_1$-norm constraint with an $\ell_p$-norm ($p>1$) constraint. After that, a radius-incorporated MK-ELM algorithm which incorporates the radius of the minimum enclosing ball (MEB) is introduced. Three efficient optimization algorithms are proposed to solve the corresponding kernel learning problems. Comprehensive experiments have been conducted on Protein, Oxford Flower17, Caltech101 and Alzheimer's disease data sets to evaluate the performance of the proposed algorithms in terms of classification accuracy and computational efficiency. As the experimental results indicate, our proposed algorithms can achieve comparable or even better classification performance than state-of-the-art MKL algorithms, while incurring much less computational cost.



## 1. Introduction

Extreme learning machine (ELM) was first designed for single hidden layer feedforward neural networks [1–3] and then extended to generalized single hidden layer feedforward networks (SLFN) which did not necessarily resemble neurons [4,5]. Different from traditional neural SLFN learning algorithms, ELM aims to minimize both training error and the norm of output weights [3,6]. Due to its (1) *high efficiency*, (2) *easy-implementation*, (3) *unification of classification and regression* and (4) *unification of binary and multi-class classification* [6], ELM has been an active research topic over the past a few years [3,6–12]. In addition, the ELM has also been successfully applied to many applications such as imbalance learning [13], missing data learning [14] and activity recognition [15], to name just a few. More recent advances in ELM can be found in [10–12].

Although researchers have made great progress from both a theoretical and a practical point of view, ELM has still not well considered the following two issues. The first one is how to choose an optimal kernel for a specific application when the kernel trick is applied to ELM such as in previous work [6,16–18]. The other one is how to handle information fusion in ELM when multiple heterogenous data sources are available. In this paper, we propose a general framework by borrowing the idea of multiple kernel learning (MKL) to handle the above two issues. We call our framework a multiple kernel extreme learning machine (MK-ELM). In the MK-ELM, the optimal kernel is assumed to be a linear combination of a group of base kernels, and the base kernel combination weights and structural parameters of ELM are jointly optimized in the learning process. Though sharing the same assumption that the optimal kernel is a linear combination of base kernels, the proposed MK-ELM and the widely studied SVM based MKL algorithms have important differences. (1) For the proposed MK-ELM, the binary and multi-class classification problems are unified into one common formula. In contrast, the one-against-one (OAO) and one-against-all (OAA) strategies are usually

* Corresponding author.
*E-mail address:* 1022xinwang.liu@gmail.com (X. Liu).

http://dx.doi.org/10.1016/j.neucom.2013.09.072
0925-2312/

adopted in SVM based MKL algorithms [19,20] to handle the multi-class classification problems. (2) The optimization problem for MK-ELM is much simpler than the one used in SVM based MKL algorithms. The structural parameter of MK-ELM can be analytically obtained by a matrix inverse operation, while a constrained quadratic programming (QP) solver is required to solve the optimization problems of SVM based MKL algorithms.

In the literature, there are mainly three research directions for existing SVM based MKL algorithms, including sparse MKL algorithms [19,21–23], non-sparse MKL algorithms [20,24] and the recent radius-incorporated MKL variants [25–27]. In order to conduct a comprehensive comparison with SVM based MKL algorithms, we also design sparse, non-sparse and radius-incorporated MK-ELM algorithms in this paper. Specifically, the contributions of this paper are highlighted as follows:

1. A sparse MK-ELM algorithm is first developed, where an $\ell_1$-norm constraint is imposed on the base kernel combination weights.
2. A non-sparse variant is proposed by substituting the $\ell_1$-norm constraint with an $\ell_p$-norm constraint, where $p>1$.
3. Another radius-incorporated MK-ELM is then proposed by integrating the radius of minimum enclosing ball (MEB) [28,29] into the objective function of MK-ELM.
4. Comprehensive experiments have been conducted to compare the proposed MK-ELM variants with existing state-of-the-art MKL algorithms, including multiple kernel SVM (MK-SVM) [19], multiple kernel least square SVM (MK-LSSVM) [30], multiple kernel fisher discriminative analysis (MK-FDA) [23], and their sparse and non-sparse variants. The experimental results demonstrate that the proposed MK-ELM variants achieve statistically comparable or better classification performance while requiring less training time.

The rest of this paper is organized as follows. We review the extreme learning machine and multiple kernel learning in Section 2. In Section 3, we first present the formulation of the sparse MK-ELM, extend it to a non-sparse case and then propose a radius-incorporated variant. Three efficient algorithms are given to solve the resulting optimization problems. Extensive experimental comparison is conducted in Sections 4 and 5 draws our conclusion.

## 2. Related work

In this section, we give a brief review of extreme learning machine and multiple kernel learning. Though ELM unifies classification and regression tasks, we only focus on classification in the following parts.

### 2.1. Extreme learning machine

According to the ELM theory [6,10], ELM aims to simultaneously minimize the training errors and the norm of output weights. This objective function, for both binary and multi-class classification tasks, can be expressed as follows:

$$\min_{\boldsymbol{\beta},\boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{\beta}\|_{\mathbf{F}}^2+\frac{C}{2}\sum_{i=1}^{n}\|\boldsymbol{\xi}_{\cdot i}\|^2 \quad \text{s.t. } \boldsymbol{\beta}^\top\boldsymbol{\phi}(\mathbf{x}_i)=\mathbf{y}_i-\boldsymbol{\xi}_{\cdot i}, \quad \forall i, \tag{1}$$

where $\{(\mathbf{x}_i,\mathbf{y}_i)\}_{i=1}^n$ is a training set, $\boldsymbol{\phi}(\mathbf{x}_i)(i=1,\ldots,n)$ is the hidden-layer output (feature mapping) corresponding to $\mathbf{x}_i$, $\boldsymbol{\beta}\in\mathbb{R}^{|\boldsymbol{\phi}(\cdot)|\times T}$ is the output weights, $\boldsymbol{\xi}\in\mathbb{R}^{T\times n}$ is the training error matrix on training data, $\boldsymbol{\xi}_{\cdot i}=[\xi_{1i},\xi_{2i},\ldots,\xi_{Ti}]^\top(1\le i\le n)$ is the $i$th column of $\boldsymbol{\xi}$, $\mathbf{y}_i=[0,\ldots,0,\underset{t}{1},0,\ldots,0]^\top\in\{0,1\}^T$ if $\mathbf{x}_i$ belongs to the $t$th ($1\le t\le T$) class, $n$ and $T$ are the number of training samples and classes, and $C$ is a regularization parameter which trades off the norm of output weights and training errors. $\|\cdot\|_{\mathbf{F}}$ is the Frobenius norm.

The optimization problem in Eq. (1) can be efficiently solved. According to [6], the optimal $\boldsymbol{\beta}^\star$ which minimizes Eq. (1) can be analytically obtained as

$$\boldsymbol{\beta}^\star=\boldsymbol{\Phi}^\top\left(\frac{\mathbf{I}}{C}+\boldsymbol{\Phi}\boldsymbol{\Phi}^\top\right)^{-1}\mathbf{Y}^\top, \tag{2}$$

where $\boldsymbol{\Phi}=[\boldsymbol{\phi}(\mathbf{x}_1),\ldots,\boldsymbol{\phi}(\mathbf{x}_n)]^\top\in\mathbb{R}^{n\times|\boldsymbol{\phi}(\cdot)|}$, $\mathbf{Y}=[\mathbf{y}_1,\ldots,\mathbf{y}_n]\in\mathbb{R}^{T\times n}$ and $\mathbf{I}$ is an identity matrix.

As can be seen from the above, both the binary and multi-class classification tasks in ELM can be handled via an unified formula Eq. (1). Moreover, Eq. (1) can be analytically solved by a matrix inverse operation, while a constrained quadratic programming problem is required in SVM. This makes the ELM easy and efficient to implement due to the fact that solving a matrix inverse problem is usually much more computationally efficient than solving the same-size constrained QP problem. In addition, it is worth mentioning that though both ELM and least square SVM (LSSVM) [31] share the same objective function as far as the optimization is concerned, there is no bias term deployed in ELM, as in Eq. (1). Such a subtle difference makes ELM to have milder optimization constraint than LSSVM. These advantages help ELM to achieve better classification performance while incurring less computational cost, as demonstrated by the experimental results in [6].

After obtaining the optimal $\boldsymbol{\beta}^\star$, the decision score of the ELM on test point $\mathbf{x}$ is determined by

$$f(\mathbf{x})=\boldsymbol{\beta}^{\star\top}\boldsymbol{\phi}(\mathbf{x}), \tag{3}$$

and the index corresponding to the highest value of $f(\mathbf{x})\in\mathbb{R}^T$ is considered as the label of $\mathbf{x}$.

### 2.2. Multiple kernel learning

It is well known that the choice of kernels is crucial for kernel-based algorithms [32]. Much effort has been devoted to tuning an optimal kernel for a specific application [19,33,27]. MKL provides an elegant way to handle such an issue by optimizing a data-dependent kernel. In MKL, the optimal kernel is assumed to be a linear combination of a group of base kernels, and the optimal combination coefficients and the structural parameters of classifiers are jointly learned by maximizing the margin [19,30], class separability criterion [24,23], etc. Specifically, MKL takes the form of

$$\kappa(\cdot,\cdot;\boldsymbol{\gamma})=\sum_{p=1}^{m}\gamma_p\kappa_p(\cdot,\cdot), \tag{4}$$

where $\{\kappa_p(\cdot,\cdot)\}_{p=1}^m$ are $m$ pre-defined base kernels, and $\{\gamma_p\}_{p=1}^m$ are the base kernel combination coefficients. Eq. (4) can be equivalently rewritten as

$$\boldsymbol{\phi}(\cdot;\boldsymbol{\gamma})=[\sqrt{\gamma_1}\boldsymbol{\phi}_1(\cdot),\sqrt{\gamma_2}\boldsymbol{\phi}_2(\cdot),\ldots,\sqrt{\gamma_m}\boldsymbol{\phi}_m(\cdot)], \tag{5}$$

where $\boldsymbol{\phi}(\cdot;\boldsymbol{\gamma})$ and $\{\boldsymbol{\phi}_p(\cdot)\}_{p=1}^m$ are the feature mappings corresponding to kernels $\kappa(\cdot,\cdot;\boldsymbol{\gamma})$ and $\{\kappa_p(\cdot,\cdot)\}_{p=1}^m$, respectively.

Usually, a constraint is imposed on the kernel combination weights $\boldsymbol{\gamma}$ to make the optimization problems bounded and the combined kernel be positive semi-definite (PSD). One common example is imposing an $\ell_q(q=1)$ norm and non-negative constraint on the kernel combination weights. Such constraint will induce sparse kernel combination, as shown in [19,21–23]. Another one is imposing an $\ell_q(q>1)$ norm and non-negative constraint. Unlike the previous one, this constraint will bring forth non-sparse kernel combination [20,24]. In the following section, we will design sparse and non-sparse multiple kernel learning algorithms for ELM by varying $q$ from one to any positive number larger than one.

## 3. Multiple kernel extreme learning machine

In [6], a kernel ELM is first proposed, in which a Gaussian kernel and a polynomial kernel are empirically specified. Such specified kernels may not be suitable for various applications. This motivates us to design a learning algorithm which is able to automatically learn a data-dependent optimal kernel for ELM in different applications. Inspired by the idea of MKL, we assume that the optimal kernel can be expressed as a linear combination of base kernels, and jointly learn the structural parameters of ELM and the optimal kernel combination coefficients. This extension makes ELM able to handle different heterogeneous data integrations, and this extends the ELM to a wider range of applications. Following the research on SVM based MKL, we first design a sparse MK-ELM algorithm, and generalize it to the non-sparse case. After that, a radius-incorporated variant is proposed. Three efficient algorithms are given to solve the corresponding kernel learning problems.

### 3.1. The sparse MK-ELM

By incorporating the base kernel combination weights into ELM, and imposing an $\ell_1$-norm and non-negative constraint on the base kernel weights, we obtain the objective function of the proposed sparse MK-ELM as follows:

$$\min_{\gamma}\min_{\boldsymbol{\beta},\boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{\beta}\|_{\mathbf{F}}^2+\frac{C}{2}\sum_{i=1}^{n}\|\boldsymbol{\xi}_{\cdot i}\|^2$$
$$\text{s.t.} \quad \boldsymbol{\beta}^{\top}\phi(\mathbf{x}_i;\boldsymbol{\gamma})=\mathbf{y}_i-\boldsymbol{\xi}_{\cdot i}, \quad \forall i, \qquad \sum_{p=1}^{m}\gamma_p=1,\ \gamma_p\geq 0, \quad \forall p, \tag{6}$$

where $\boldsymbol{\beta}=[\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_m]\in\mathbb{R}^{(|\phi_1(\cdot)|+\cdots+|\phi_m(\cdot)|)\times T}$, $\boldsymbol{\beta}_p\in\mathbb{R}^{|\phi_p(\cdot)|\times T}(p=1,\ldots,m)$ is the $p$th component corresponding to the $p$th base kernel. Recall that $\boldsymbol{\xi}\in\mathbb{R}^{T\times n}$ is the training error matrix on training data, $\boldsymbol{\xi}_{\cdot i}=[\xi_{1i},\xi_{2i},\ldots,\xi_{Ti}]^{\top}(1\leq i\leq n)$ is the $i$th column of $\boldsymbol{\xi}$.

As observed, Eq. (6) optimizes the structural parameter of ELM $\boldsymbol{\beta}$ and the kernel combination coefficients $\boldsymbol{\gamma}$ jointly. We now show how to solve the objective function in Eq. (6) efficiently. By substituting Eq. (5) into Eq. (6), Eq. (6) can be rewritten as

$$\min_{\gamma}\min_{\boldsymbol{\beta},\boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{\beta}\|_{\mathbf{F}}^2+\frac{C}{2}\sum_{i=1}^{n}\|\boldsymbol{\xi}_{\cdot i}\|^2$$
$$\text{s.t.} \quad \sum_{p=1}^{m}\sqrt{\gamma_p}\boldsymbol{\beta}_p^{\top}\phi_p(\mathbf{x}_i)=\mathbf{y}_i-\boldsymbol{\xi}_{\cdot i}, \quad \forall i, \qquad \sum_{p=1}^{m}\gamma_p=1,\ \gamma_p\geq 0, \quad \forall p. \tag{7}$$

After defining

$$\tilde{\boldsymbol{\beta}}=[\tilde{\boldsymbol{\beta}}_1,\tilde{\boldsymbol{\beta}}_2,\ldots,\tilde{\boldsymbol{\beta}}_m], \tag{8}$$

where $\tilde{\boldsymbol{\beta}}_p=\sqrt{\gamma_p}\boldsymbol{\beta}_p, p=1,\ldots,m$, Eq. (7) can be equivalently reformulated as

$$\min_{\gamma}\min_{\tilde{\boldsymbol{\beta}},\boldsymbol{\xi}} \quad \frac{1}{2}\sum_{p=1}^{m}\frac{\|\tilde{\boldsymbol{\beta}}_p\|_{\mathbf{F}}^2}{\gamma_p}+\frac{C}{2}\sum_{i=1}^{n}\|\boldsymbol{\xi}_{\cdot i}\|^2$$
$$\text{s.t.} \quad \sum_{p=1}^{m}\tilde{\boldsymbol{\beta}}_p^{\top}\phi_p(\mathbf{x}_i)=\mathbf{y}_i-\boldsymbol{\xi}_{\cdot i}, \quad \forall i, \qquad \sum_{p=1}^{m}\gamma_p=1,\ \gamma_p\geq 0, \quad \forall p. \tag{9}$$

It is not difficult to verify that Eq. (9) is a joint-convex optimization problem [34], and its Lagrangian function is

$$L(\tilde{\boldsymbol{\beta}},\boldsymbol{\xi},\boldsymbol{\gamma})=\frac{1}{2}\sum_{p=1}^{m}\frac{\|\tilde{\boldsymbol{\beta}}_p\|_{\mathbf{F}}^2}{\gamma_p}+\frac{C}{2}\sum_{i=1}^{n}\|\boldsymbol{\xi}_{\cdot i}\|^2 - \sum_{t=1}^{T}\sum_{i=1}^{n}\alpha_{it}\left(\sum_{p=1}^{m}\tilde{\boldsymbol{\beta}}_p^{\top}\phi_p(\mathbf{x}_i)-y_{ti}+\xi_{ti}\right)+\tau\left(\sum_{p=1}^{m}\gamma_p-1\right), \tag{10}$$

where $\boldsymbol{\alpha}\in\mathbb{R}^{n\times T}$ and $\tau$ are the Lagrange multipliers. In Eq. (10), we omit the non-negative constraints on $\gamma_p,(p=1,\ldots,m)$ since the newly updated kernel combination weights are automatically kept non-negative at each iteration, as will be validated later.

We can have the KKT optimality conditions of Eq. (10) as follows:

$$\tilde{\boldsymbol{\beta}}_p=\gamma_p\sum_{t=1}^{T}\sum_{i=1}^{n}\alpha_{it}\phi_p(\mathbf{x}_i), \quad \forall p \tag{11}$$

$$\xi_{ti}=\frac{\alpha_{ti}}{C}, \quad \forall t \ \forall i \tag{12}$$

$$\sum_{p=1}^{m}\tilde{\boldsymbol{\beta}}_p^{\top}\phi_p(\mathbf{x}_i)=\mathbf{y}_i-\boldsymbol{\xi}_{\cdot i}, \quad \forall i, \tag{13}$$

which can be rewritten into a matrix form as

$$\left(\mathbf{K}(\cdot,\cdot;\boldsymbol{\gamma})+\frac{\mathbf{I}}{C}\right)\boldsymbol{\alpha}=\mathbf{Y}^{\top}, \tag{14}$$

where $\mathbf{K}(\mathbf{x}_i,\mathbf{x}_j;\boldsymbol{\gamma})=\phi(\mathbf{x}_i;\boldsymbol{\gamma})^{\top}\phi(\mathbf{x}_j;\boldsymbol{\gamma})=\sum_{p=1}^{m}\gamma_p\mathbf{K}_p(\mathbf{x}_i,\mathbf{x}_j)$. $\mathbf{Y}=[\mathbf{y}_1,\ldots,\mathbf{y}_n]\in\mathbb{R}^{T\times n}$ is the label matrix. From Eq. (14), the $\boldsymbol{\alpha}$, which corresponds to the structural parameter of ELM, can be obtained by

$$\boldsymbol{\alpha}=\left(\mathbf{K}(\cdot,\cdot;\boldsymbol{\gamma})+\frac{\mathbf{I}}{C}\right)^{-1}\mathbf{Y}^{\top}. \tag{15}$$

We then show how to update the kernel combination coefficients $\boldsymbol{\gamma}$ efficiently. By taking the derivative of Eq. (10) w.r.t $\gamma_p(p=1,\ldots,m)$ and let it vanish, we obtain that the new kernel combination weights $\boldsymbol{\gamma}^{new}$ are updated by

$$\gamma_p^{new}=\frac{\|\tilde{\boldsymbol{\beta}}_p\|_{\mathbf{F}}}{\sum_{p=1}^{m}\|\tilde{\boldsymbol{\beta}}_p\|_{\mathbf{F}}}, \quad \forall p, \tag{16}$$

where

$$\|\tilde{\boldsymbol{\beta}}_p\|_{\mathbf{F}}=\gamma_p\sqrt{\sum_{s,t=1}^{T}\sum_{i,j=1}^{n}\alpha_{it}\alpha_{js}\mathbf{K}_p(\mathbf{x}_i,\mathbf{x}_j)}. \tag{17}$$

and $\gamma_p,(p=1,\ldots,m)$ is the $p$th kernel combination weight in the last iteration. As seen from Eqs. (16) and (17), the newly updated $\gamma_p^{new}(p=1,\ldots,m)$ are kept non-negative at each iteration, which automatically satisfies the non-negative constraint. The detailed derivation of updating the kernel combination weights is provided in the Appendix.

The overall optimization algorithm for solving sparse MK-ELM is presented in Table 1.

**Algorithm 1.** The sparse MK-ELM.

1: **Input**: $\{\mathbf{K}_p\}_{p=1}^{m}$, $\mathbf{y}$ and $C$.
2: **Output**: $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$.
3: Initialize $\boldsymbol{\gamma}=\boldsymbol{\gamma}^0$ and $t=0$.
4: **repeat**
5: Compute $\mathbf{K}(\cdot,\cdot;\boldsymbol{\gamma})=\sum_{p=1}^{m}\gamma_p^t\mathbf{K}_p$.
6: Update $\boldsymbol{\alpha}^t$ by solving Eq. (15).
7: Update $\boldsymbol{\gamma}^{t+1}$ by Eq. (16).
8: $t=t+1$.
9: **until** $\max\{|\boldsymbol{\gamma}^{t+1}-\boldsymbol{\gamma}^t|\}\leq 1e-4$

### 3.2. Non-sparse MK-ELM

Recent research on SVM based MKL has indicated that non-sparse MKL algorithms can usually outperform the sparse alternatives [20,35] by arguing that some complementary information may be lost due to the sparsity constraint. In the following part, we first design a non-sparse MK-ELM algorithm, and propose an optimization algorithm to solve the resulting kernel learning

problem efficiently. Specifically, the objective function for non-sparse MK-ELM can be reformulated as

$$\min_{\gamma}\min_{\beta,\xi}\ \frac{1}{2}\|\boldsymbol{\beta}\|_{\mathbf{F}}^2+\frac{C}{2}\sum_{i=1}^{n}\|\boldsymbol{\xi}_{\cdot i}\|^2$$
$$\text{s.t.}\quad \boldsymbol{\beta}^{\top}\phi(\mathbf{x}_i;\boldsymbol{\gamma})=\mathbf{y}_i-\boldsymbol{\xi}_{\cdot i},\ \ \forall i,\quad \sum_{p=1}^{m}\gamma_p^q=1,\ \gamma_p\geq 0,\ \forall p, \tag{18}$$

where $q>1$ is a scalar.

As can be seen, the objective function of non-sparse MK-ELM in Eq. (18) is almost the same as the one defined in Eq. (6), with only one important difference: an $\ell_q$-norm ($q>1$) is imposed on $\boldsymbol{\gamma}$, leading to a non-sparse kernel combination. The algorithm for solving Eq. (18) is similar to Eq. (6), except for the way in which the kernel combination weights are updated. Specifically, the kernel combination weights for non-sparse MK-ELM are updated by Eq. (19)

$$\gamma_p=\frac{\|\tilde{\boldsymbol{\beta}_p}\|_{\mathbf{F}}^{2/(1+q)}}{(\sum_{p=1}^{m}\|\tilde{\boldsymbol{\beta}_p}\|_{\mathbf{F}}^{2q/(1+q)})^{1/q}},\quad \forall p. \tag{19}$$

The detailed derivations are provided in the Appendix. The overall algorithm for solving the non-sparse MK-ELM is presented in Algorithm 2.

**Algorithm 2.** The non-sparse MK-ELM.

1: **Input**: $\{\mathbf{K}_p\}_{p=1}^m$, $\mathbf{y}$, $q$ and $C$.
2: **Output**: $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$.
3: Initialize $\boldsymbol{\gamma}=\boldsymbol{\gamma}^0$ and $t=0$.
4: **repeat**
5: Compute $\mathbf{K}(\cdot,\cdot;\boldsymbol{\gamma})=\sum_{p=1}^{m}\gamma_p^t\mathbf{K}_p$.
6: Update $\boldsymbol{\alpha}^t$ by solving Eq. (15).
7: Update $\boldsymbol{\gamma}^{t+1}$ by Eq. (19).
8: $t=t+1$.
9: **until** $\max\{|\boldsymbol{\gamma}^{t+1}-\boldsymbol{\gamma}^t|\}\leq 1e-4$

### 3.3. Radius-incorporated MK-ELM

More recent research on MKL has demonstrated that the performance of traditional MKL algorithms [21,19], which maximize the margin only, could be further improved by incorporating radius information [25–27]. In the following, we propose a simple but effective approach to incorporate the radius information into MK-ELM. Inspired by the work in [25], we integrate the radius of minimum enclosing ball (MEB) via one of its upper bound, i.e., a linear combination of base radiuses corresponding to each base-kernel-induced feature space. Specifically, we approximate the squared radius of MEB with $\sum_{p=1}^{m}\gamma_p R_p^2$, where $R_p$ is the radius of MEB in $\kappa_p$-induced feature space. The benefits of such an approximation are two-fold: (1) it leads to a joint-convex optimization problem, despite incorporating radius information into MK-ELM. (2) It makes the resulting optimization problem easily implemented via a minor modification of the codes of sparse MK-ELM.

The objective function of the radius-incorporated MK-ELM is presented in the following equation:

$$\min_{\gamma}\min_{\tilde{\beta},\xi}\ \frac{1}{2}\left(\sum_{p=1}^{m}\gamma_p R_p^2\right)\sum_{p=1}^{m}\frac{\|\tilde{\boldsymbol{\beta}_p}\|_{\mathbf{F}}^2}{\gamma_p}+\frac{C}{2}\sum_{i=1}^{n}\|\boldsymbol{\xi}_{\cdot i}\|^2$$
$$\text{s.t.}\quad \sum_{p=1}^{m}\tilde{\boldsymbol{\beta}_p}^{\top}\phi_p(\mathbf{x}_i)=\mathbf{y}_i-\boldsymbol{\xi}_{\cdot i},\ \ \forall i,\ \gamma_p\geq 0,\ \forall p. \tag{20}$$

As can be seen, the incorporation of radius enables the formulation in Eq. (20) to handle the scaling issues as pointed out in [26] and thus, it is a bounded optimization problem. Therefore, it is not necessary to impose any norm constraint on the kernel combination weights. However, the $\ell_1$-norm and $\ell_q(q>1)$−norm constraints are required in sparse MK-ELM (see Eq. (6)) and non-sparse MK-ELM (Eq. (18)) respectively to make the corresponding optimization problems bounded. We now theoretically show that the optimization problem in Eq. (20) can be reformulated as one similar to sparse MK-ELM, with only one crucial difference that a base radius weighted norm constraint is imposed on the kernel combination weights. By this way, the radius information is encoded, as stated in Theorem 2 (the proof can be found in the Appendix).

**Theorem 1.** *Eq. (20) can be equivalently addressed by solving the following optimization problem in Eq. (21):*

$$\min_{\gamma}\min_{\tilde{\beta},\xi}\ \frac{1}{2}\sum_{p=1}^{m}\frac{\|\tilde{\boldsymbol{\beta}_p}\|_{\mathbf{F}}^2}{\gamma_p}+\frac{C}{2}\sum_{i=1}^{n}\|\boldsymbol{\xi}_{\cdot i}\|^2$$
$$\text{s.t.}\quad \sum_{p=1}^{m}\tilde{\boldsymbol{\beta}_p}^{\top}\phi_p(\mathbf{x}_i)=\mathbf{y}_i-\boldsymbol{\xi}_{\cdot i},\ \ \forall i,\quad \sum_{p=1}^{m}\gamma_p R_p^2=1,\gamma_p\geq 0,\forall p. \tag{21}$$

As can be observed, the optimization problem in Eq. (21) is a joint-convex one and its Lagrangian function of Eq. (21) is as follows:

$$L(\tilde{\boldsymbol{\beta}},\boldsymbol{\xi},\boldsymbol{\gamma})=\frac{1}{2}\sum_{p=1}^{m}\frac{\|\tilde{\boldsymbol{\beta}_p}\|_{\mathbf{F}}^2}{\gamma_p}+\frac{C}{2}\sum_{i=1}^{n}\|\boldsymbol{\xi}_{\cdot i}\|^2$$
$$-\sum_{t=1}^{T}\sum_{i=1}^{n}\alpha_{it}\left(\sum_{p=1}^{m}\tilde{\boldsymbol{\beta}_p}^{\top}\phi_p(\mathbf{x}_i)-y_{ti}+\xi_{ti}\right)+\tau\left(\sum_{p=1}^{m}\gamma_p R_p^2-1\right), \tag{22}$$

where $\boldsymbol{\alpha}$ and $\tau$ are Lagrange multipliers. We omit the non-negative constraints on $\boldsymbol{\gamma}$ due to the fact that the non-negativity of $\boldsymbol{\gamma}$ will be automatically kept at each iteration in the optimization process.

Similarly, the Lagrange multipliers $\boldsymbol{\alpha}$, which corresponds to the structural parameter of radius-incorporated MK-ELM in Eq. (21), can be obtained by solving Eq. (15). By taking the derivative of Eq. (22) with respect to $\gamma_p,(p=1,\ldots,m)$ and letting it vanish, we update the optimal $\boldsymbol{\gamma}$ by

$$\gamma_p=\frac{\|\tilde{\boldsymbol{\beta}_p}\|_{\mathbf{F}}}{R_p\sum_{p=1}^{m}R_p\|\tilde{\boldsymbol{\beta}_p}\|_{\mathbf{F}}},\ \forall p, \tag{23}$$

where $\|\tilde{\boldsymbol{\beta}_p}\|_{\mathbf{F}}$ is defined as in Eq. (17). Note that the $\|\tilde{\boldsymbol{\beta}_p}\|_{\mathbf{F}}(p=1,\ldots,m)$ is weighted by $R_p(p=1,\ldots,m)$ in radius-incorporated MK-ELM, which is different from the sparse and non-sparse MK-ELM, where all $\|\tilde{\boldsymbol{\beta}_p}\|_{\mathbf{F}}(p=1,\ldots,m)$ are treated equally (see Eqs. (16) and (19)). The overall optimization algorithm for solving the radius-incorporated MK-ELM is presented in Algorithm 3.

**Algorithm 3.** The radius-incorporated MK-ELM.

1: **Input**: $\{\mathbf{K}_p\}_{p=1}^m$, $\mathbf{y}$ and $C$.
2: **Output**: $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$.
3: Initialize $\boldsymbol{\gamma}=\boldsymbol{\gamma}^0$ and $t=0$.
4: Calculate $R_p^2(p=1,\ldots,m)$ with $\mathbf{K}_p(\forall p)$.
5: **repeat**
6: Compute $\mathbf{K}(\cdot,\cdot;\boldsymbol{\gamma})=\sum_{p=1}^{m}\gamma_p^t\mathbf{K}_p$.
7: Update $\boldsymbol{\alpha}^t$ by solving Eq. (15).
8: Update $\boldsymbol{\gamma}^{t+1}$ by Eq. (23).
9: $t=t+1$.
10: **until** $\max\{|\boldsymbol{\gamma}^{t+1}-\boldsymbol{\gamma}^t|\}\leq 1e-4$

After we obtain the optimal $\boldsymbol{\alpha}^\star$ and $\boldsymbol{\gamma}^\star$ by Algorithms 1, 2 or 3, the decision score of sparse (non-sparse and radius-incorporated) MK-ELM on a sample $\mathbf{x}$ can be expressed as

$$f(\mathbf{x})=[f_1(\mathbf{x}),\ldots,f_T(\mathbf{x})], \tag{24}$$

where

$$f_c(\mathbf{x})=\sum_{i=1}^{n}\alpha_{ic}^{\star}\sum_{p=1}^{m}\gamma_p^{\star}\mathbf{K}_p(\mathbf{x}_i,\mathbf{x}),\quad ,1\leq c\leq T. \tag{25}$$

Finally, the index corresponding to the highest value of $f(\mathbf{x})$ is considered as the label of $\mathbf{x}$.

### 3.4. Discussion

We end this section by discussing the differences between our proposed MK-ELM and SVM based MKL algorithms [19,20] and LSSVM based MKL algorithms [30].

The differences between the proposed MK-ELM and SVM based MKL algorithms [19,20] can be summarized as follows: (1) the strategies in handling multi-class classification tasks. The objective function of the proposed MK-ELM is directly designed for multi-class classification problems, while the OAO or OAA strategy is used in [19,20]. This difference allows MK-ELM to achieve better classification performance than SVM based MKL algorithms. (2) Computational efficiency. For the proposed MK-ELM, a matrix inverse operation is involved at each iteration, while $T$ ($T$ is the number of classes) constrained quadratic programming (QP) problems are required to be solved at each iteration when using OAA strategy for SVM based MKL algorithms [19,20]. Such difference makes the proposed MK-ELM much more computationally efficient than SVM based MKL algorithms at each iteration. (3) Easy-implementation. Our proposed MK-ELM algorithms are easy to implement via a matrix inverse operation. In contrast, a constrained QP solver is needed in SVM based MKL algorithms [19,20].

Now we highlight the differences between the proposed MK-ELM and LSSVM based MKL algorithms [30] as follows: (1) milder optimization constraints. Though the proposed MK-ELM and LSSVM based MKL algorithms have the same objective function, there is no bias term used in MK-ELM (see the constraints in Eq. (6) for the details), while the bias term is applied in [30]. The application of this bias term incurs additional equality constraints on the dual problem of LSSVM based MKL algorithms. However, MK-ELM does not have these additional constraints. This makes the proposed MK-ELM have milder optimization constraints than LSSVM based MKL algorithms [30], and thus, compared to the optimization problem of MK-ELM, the solution obtained by LSSVM based MKL algorithms is suboptimal [6]. (2) The strategies in handling multi-class classification tasks are different. Again, these two algorithms differ in handling multi-class classification tasks, as mentioned above.

## 4. Experimental results

### 4.1. Experimental settings

Our experiment is composed of two parts. The first part compares the proposed sparse MK-ELM ($\ell_1$-MK-ELM) and radius-incorporated MK-ELM with several state-of-the-art MKL algorithms, including $\ell_1$-MK-SVM [19], $\ell_1$-MK-FDA [23] and $\ell_1$-MK-LSSVM [30]. The best results achieved by each single kernel of these algorithms are also reported as a reference. The second part evaluates the proposed non-sparse MK-ELM against the non-sparse MK-SVM [20] and non-sparse MK-FDA [24]. We implement the proposed sparse MK-ELM, radius-incorporated MK-ELM and non-sparse MK-ELM by ourselves, while the codes for other algorithms are downloaded from the authors' web sites.[1,2,3,4,5]

We compare the performance of the above-mentioned algorithms based on (1) classification performance, including both classification accuracy (ACC) and the widely used mean average precision (mAP), and (2) learning time cost (including cross-validation and training time), to demonstrate both the effectiveness and the computational efficiency of the proposed MK-ELM algorithms. In our experiments, the regularization parameter for each algorithm is chosen from an appropriately large $[10^{-3}, 10^{-1}, \ldots, 10^{4}]$ by its performance on a separated validation set or 5-fold cross-validation on the training data. All experiments are conducted on a high performance cluster server, where each node has 2.3 GHz CPU and 16 GB memory.

### 4.2. Data sets

The aforementioned MKL algorithms are evaluated on three benchmark MKL data sets and a real world application data set, including the protein fold prediction data set,[6] the Oxford Flower17 data set,[7] the Caltech101 data set[8] and Alzheimer's disease data set [36].

Protein fold prediction is a multi-source and multi-class data set based on a subset of the PDB-40D SCOP collection. It contains 12 different feature spaces, including composition, secondary, hydrophobicity, volume, polarity, polarizability, L1, L4, L14, L30, SWblosum62 and SWpam50. It is a typical multiple source integration classification problem and this data set has been widely adopted in the MKL community [37–39]. Similar to the protein folding prediction, the Oxford Flowers and Caltech101 are also two multi-class benchmark data sets, which are used to evaluate the performance of MKL algorithms [40,37,38,20].

Besides the three benchmarks, we also compare the MKL algorithms on Alzheimer's disease data set [36], which uses four heterogeneous data sources, including cerebrospinal fluid (CSF) biomarkers, the left hippocampal shapes, the right hippocampal shapes and the brain regional gray matter volumes. In detail, each datum is represented by 229 features, which include three CSF biomarkers, 63 left hippocampal shape features, 63 right hippocampal shape features, and 100 regional gray matter volumes. The above MKL algorithms are applied to differentiate different clinical groups based on their baseline scanning, including mild cognitive impairment (MCI) versus normal controls (NC), prodromal mild cognitive impairment (PMCI) versus NC, and PMCI versus stable mild cognitive impairment (SMCI), as detailed in the lower portion of Table 1.

For Caltech101, the base kernel matrices have been pre-computed [39]. For protein, Flower17 and Alzheimer's disease data sets, the original data or dissimilar matrices are used to generate the base kernel matrices as follows. For each data source or dissimilar matrix, four types of kernels are applied: the Gaussian kernel (i.e., $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma)$), Laplacian kernel (i.e., $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|/\sqrt{\sigma})$), inverse square distance kernel (i.e., $k(\mathbf{x}_i, \mathbf{x}_j) = 1/((\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma)+1)$), and inverse distance kernel (i.e., $k(\mathbf{x}_i, \mathbf{x}_j) = 1/((\|\mathbf{x}_i - \mathbf{x}_j\|/\sqrt{\sigma})+1)$), where $\sigma$ is the kernel parameter. They represent different ways to utilize the Euclidean distance between two samples. In our experiments, five kernel parameters $2^t\sigma_0(t \in \{-2, -1, 0, 1, 2\})$ are employed for each type of kernel, where $\sigma_0$ is set to be the averaged pairwise Euclidean distance between samples. In this way, we generate $\sharp(\text{src}) \times 4 \times 5$ base kernels and use them for all the MKL algorithms compared in our experiment, where $\sharp(\text{src})$ is the number of data sources. The detailed information about these data sets is presented in Table 1.

Following the same setting in [24], the experiments on protein, Flower17 and Caltech101 are repeated 10, 10 and three times, respectively. For Alzheimer's disease data sets, we repeat the

[1] http://asi.insa-rouen.fr/enseignants/~arakoto/code/mklindex.html
[2] http://www.public.asu.edu/~jye02/Software/DKL/index.htm
[3] http://homes.esat.kuleuven.be/~sistawww/bioi/syu/l2lssvm.html
[4] http://doc.ml.tu-berlin.de/nonsparse_mkl/
[5] http://kahlan.eps.surrey.ac.uk/featurespace/web/mkl/lp_mk_fda.html

[6] http://mkl.ucsd.edu/dataset/protein-fold-prediction
[7] http://www.robots.ox.ac.uk/~vgg/data/flowers/17/index.html
[8] http://kahlan.eps.surrey.ac.uk/featurespace/web/mkl/lp_mk_fda.html

experiments 100 times due to the scarcity of samples. The mean accuracy and standard deviation are reported. To conduct a rigorous comparison, the *paired Student's t-test* is performed. The $p$-value of the pairwise $t$-test represents the probability that two sets of compared results come from distributions with an equal mean. A $p$-value of 0.05 is considered statistically significant.

### 4.3. Comparison of sparse MKL algorithms

#### 4.3.1. Results on protein fold prediction

Table 2 reports the results of sparse MKL on protein fold prediction data set, where the upper and lower parts correspond to classification accuracy (ACC) and mean average precision (mAP), respectively.

For the upper part of Table 2, each cell has four elements which represent mean accuracy, standard deviation, $p$-value obtained by statistical test and the training time. We have the following observations:

1. The proposed $\ell_1$-MK-ELM and $\ell_1$-MK-LSSVM [30] achieve the best classification accuracy, which is much better than $\ell_1$-MK-SVM [19] and $\ell_1$-MK-FDA [23]. Moreover, the radius-incorporated MK-ELM further improves the classification accuracy of $\ell_1$-MK-ELM, indicating the effectiveness of incorporating the radius information.
2. The proposed radius-incorporated MK-ELM obtains the highest mAP among all the compared MKL algorithms, where 4.32% improvement is gained over the second best one. Also, the statistical test results validate that the improvement is significant.
3. The computational cost of the proposed $\ell_1$-MK-ELM and the radius-incorporated MK-ELM is much lower than that of MK-LSSVM [30]. Specifically, it takes MK-LSSVM [30] 8.24$e$+003 s to finish the training procedure, which is approximately three and two times longer than that of $\ell_1$-MK-ELM and radius-incorporated MK-ELM, respectively.

**Table 1**
Information of the data used in our experiments.

| Data | Number of training | Validation | Testing | Classes | Sources |
|---|---|---|---|---|---|
| Protein | 248 | 63 | 383 | 27 | 12 |
| Flower17 | 680 | 340 | 340 | 17 | 7 |
| Caltech101 | 1010 | 505 | 3999 | 101 | 10 |
| PMCI vs. NC | 72 | – | 48 | 2 | 4 |
| MCI vs. NC | 115 | – | 76 | 2 | 4 |
| PMCI vs. SMCI | 73 | – | 48 | 2 | 4 |

#### 4.3.2. Results on Oxford Flower17

As seen in Table 3, $\ell_1$-MK-ELM achieves a slight improvement over MK-LSSVM [30] in terms of classification accuracy, and this improvement is further raised to 3.73% by radius-incorporated MK-ELM, again indicating the importance of incorporating the radius information. Similar results can be observed on mAP, where $\ell_1$-MK-ELM and the radius-incorporated MK-ELM obtain 0.94% and 3.75% improvement over the second best one, respectively. In addition, the computational efficiency is compared among the algorithms that achieve comparable classification performance. The result again shows the efficiency of the proposed algorithms, as presented in the last row of each cell.

#### 4.3.3. Results on Caltech101

Table 4 gives the results on Caltech101. From this table, we once again observe that the best performance is achieved by the proposed $\ell_1$-MK-ELM and the radius-incorporated MK-ELM. In addition, the training time of MK-LSSVM [30] is approximately five and three times longer than $\ell_1$-MK-ELM and the radius-incorporated MK-ELM, respectively.

#### 4.3.4. Results on Alzheimer's disease

The results on Alzheimer's disease data sets are reported in Table 5. We observe that our proposed algorithms achieve significant

**Table 2**
Performance comparison with statistical test on the protein fold prediction data set. The four rows of each cell represent ACC/mAP, standard derivation, $p$-value and training time (in seconds), respectively. Boldface means no statistical difference from the best one ($p$-value $\geq 0.05$).

| Criteria | MK-ELM (proposed) | | | MK-SVM [19] | | MK-FDA [23] | | MK-LSSVM [30] | |
|---|---|---|---|---|---|---|---|---|---|
| | $\ell_1$ | Single best | Radius | $\ell_1$ | Single best | $\ell_1$ | Single best | $\ell_1$ | Single best |
| ACC | **64.20** | 54.62 | **65.14** | 53.47 | 54.26 | 59.43 | 46.61 | **64.73** | 54.07 |
| | ± **1.47** | ± 3.44 | ± **1.22** | ± 1.01 | ± 3.02 | ± 1.20 | ± 3.06 | ± **2.06** | ± 3.58 |
| | **0.13** | 0.00 | **1.00** | 0.00 | 0.00 | 0.00 | 0.00 | **0.57** | 0.00 |
| | 2.88$e$+003 | 10.15 | 4.63$e$+003 | 3.72$e$+003 | 5.57$e$+003 | 575.65 | 21.80 | 8.24$e$+003 | 593.77 |
| mAP | 69.52 | 68.22 | **74.34** | 56.45 | 67.55 | 69.43 | 67.10 | 70.02 | 67.25 |
| | ± 1.33 | ± 1.40 | ± **1.42** | ± 0.97 | ± 1.56 | ± 1.23 | ± 2.04 | ± 1.27 | ± 2.03 |
| | 0.00 | 0.00 | **1.00** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

**Table 3**
Performance comparison with statistical test on Oxford Flower17 data set. The four rows of each cell represent ACC/mAP, standard derivation, $p$-value and training time (in seconds), respectively. Boldface means no statistical difference from the best one ($p$-value $\geq 0.05$).

| Criteria | MK-ELM (proposed) | | | MK-SVM [19] | | MK-FDA [23] | | MK-LSSVM [30] | |
|---|---|---|---|---|---|---|---|---|---|
| | $\ell_1$ | Single best | Radius | $\ell_1$ | Single best | $\ell_1$ | Single best | $\ell_1$ | Single best |
| ACC | **81.18** | 70.49 | **84.51** | 73.33 | 70.39 | 77.45 | 65.10 | 80.78 | 70.29 |
| | ± **1.64** | ± 0.95 | ± **2.07** | ± 2.09 | ± 0.95 | ± 0.90 | ± 0.61 | ± 1.77 | ± 1.28 |
| | **0.06** | 0.01 | **1.00** | 0.01 | 0.00 | 0.04 | 0.00 | 0.00 | 0.00 |
| | 1.10$e$+004 | 62.91 | 1.34$e$+004 | 2.90$e$+004 | 3.28$e$+004 | 1.49$e$+003 | 164.89 | 1.60$e$+004 | 0.30$e$+003 |
| mAP | 87.76 | 75.31 | **90.57** | 80.47 | 75.54 | 86.82 | 73.70 | 85.44 | 74.02 |
| | ± 1.28 | ± 0.18 | ± **1.51** | ± 1.78 | ± 0.35 | ± 1.11 | ± 0.21 | ± 2.43 | ± 2.47 |
| | 0.01 | 0.00 | **1.00** | 0.03 | 0.01 | 0.01 | 0.00 | 0.02 | 0.00 |

**Table 4**
Performance comparison with statistical test on Caltech101 data set. The four rows of each cell represent ACC/mAP, standard derivation, $p$-value and training time (in seconds), respectively. Boldface means no statistical difference from the best one ($p$-value $\geq 0.05$).

| Criteria | MK-ELM (proposed) | | | MK-SVM [19] | | MK-FDA [23] | | MK-LSSVM [30] | |
|---|---|---|---|---|---|---|---|---|---|
| | $\ell_1$ | Single best | Radius | $\ell_1$ | Single best | $\ell_1$ | Single best | $\ell_1$ | Single best |
| ACC | **64.73** | 61.22 | **65.21** | 63.78 | 61.03 | 57.13 | 55.16 | **64.57** | 61.17 |
| | **±2.11** | ±1.89 | **±1.89** | ±2.41 | ±2.40 | ±1.48 | ±0.46 | **±2.45** | ±2.28 |
| | **0.08** | 0.01 | **1.00** | 0.04 | 0.00 | 0.00 | 0.01 | **0.22** | 0.02 |
| | 8.34e+003 | 14.40 | 1.13e+004 | 1.13e+005 | 9.34e+003 | 459.48 | 37.89 | 3.74e+004 | 3.17e+003 |
| mAP | **65.92** | 62.20 | **66.38** | 64.67 | 61.65 | 63.06 | 60.07 | 65.86 | 62.38 |
| | **±2.05** | ±1.98 | **±1.96** | ±2.45 | ±2.39 | ±2.47 | ±2.32 | ±2.22 | ±2.10 |
| | **0.09** | 0.01 | **1.00** | 0.03 | 0.01 | 0.01 | 0.01 | 0.02 | 0.01 |

**Table 5**
Classification accuracy comparison with statistical test on Alzheimer's disease data sets. The four rows of each cell represent ACC/mAP, standard derivation, $p$-value and training time (in seconds), respectively. Boldface means no statistical difference from the best one ($p$-value $\geq 0.05$).

| | MK-ELM (proposed) | | | MK-SVM [19] | | MK-FDA [23] | | MK-LSSVM [30] | |
|---|---|---|---|---|---|---|---|---|---|
| | $\ell_1$ | Single best | Radius | $\ell_1$ | Single best | $\ell_1$ | Single best | $\ell_1$ | Single best |
| ACC PMCI vs. NC | 83.08 | 77.81 | **83.35** | **84.10** | 78.40 | 81.90 | 75.98 | 76.17 | 77.90 |
| | ±5.37 | ±5.76 | **±5.58** | **±5.46** | ±6.31 | ±4.96 | ±7.05 | ±5.33 | ±6.15 |
| | 0.00 | 0.00 | **0.05** | **1.00** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | 15.29 | 0.93 | 43.83 | 188.75 | 70.57 | 2.38 | 1.69 | 34.78 | 1.23 |
| ACC MCI vs. NC | **70.83** | 64.67 | **71.25** | 68.93 | 64.58 | 67.55 | 65.21 | 67.21 | 64.99 |
| | **±3.66** | ±4.81 | **±3.67** | ±3.79 | ±3.86 | ±4.22 | ±4.91 | ±4.05 | ±4.53 |
| | **0.21** | 0.00 | **1.00** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | 30.79 | 2.96 | 60.04 | 366.87 | 117.73 | 4.34 | 5.10 | 33.00 | 3.63 |
| ACC PMCI vs. SMCI | **67.37** | 62.58 | **67.54** | 66.46 | 62.29 | 65.62 | 62.29 | 65.10 | 63.00 |
| | **±6.05** | ±6.25 | **±5.81** | ±5.58 | ±6.29 | ±5.82 | ±6.32 | ±5.24 | ±5.97 |
| | **0.71** | 0.00 | **1.00** | 0.04 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | 12.71 | 0.99 | 42.59 | 193.37 | 76.13 | 2.34 | 1.83 | 18.02 | 1.31 |
| mAP PMCI vs. NC | **94.69** | 87.42 | **94.97** | **94.78** | 88.00 | 94.19 | 87.78 | 94.49 | 87.24 |
| | **±2.57** | ±4.41 | **±2.19** | **±2.51** | ±4.50 | ±2.68 | ±4.20 | ±2.11 | ±4.53 |
| | **0.06** | 0.00 | **1.00** | **0.14** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| mAP MCI vs. NC | 86.96 | 82.42 | **87.91** | **87.93** | 82.50 | **87.43** | 83.66 | **87.97** | 82.95 |
| | ±3.52 | ±4.52 | **±3.12** | **±3.10** | ±4.33 | **±3.48** | ±3.84 | **±3.03** | ±4.26 |
| | 0.01 | 0.00 | **0.87** | **0.90** | 0.00 | **0.18** | 0.00 | **1.00** | 0.00 |
| mAP PMCI vs. SMCI | **67.89** | 61.15 | **68.04** | **68.06** | 61.26 | **67.82** | 63.20 | **69.33** | 61.33 |
| | **±6.77** | ±7.63 | **±6.89** | **±7.09** | ±6.84 | **±6.62** | ±7.42 | **±7.02** | ±7.73 |
| | **0.08** | 0.00 | **0.12** | **0.17** | 0.00 | **0.08** | 0.00 | **1.00** | 0.00 |

improvement over the other algorithms in terms of classification accuracy on the tasks of MCI vs. NC and PMCI vs. SMCI. Specifically, compared with $\ell_1$-MK-SVM [19], the radius-incorporated MK-ELM improves the classification accuracy by 2.32% while only consuming one-sixth of its training time on MCI vs. NC. Also, the radius-incorporated MK-ELM achieves 4.04% improvement over $\ell_1$-MK-LSSVM [30] although taking twice the computational cost. Similar conclusions can be drawn for PMCI vs. SMCI.

Overall, we can draw the following conclusions from the above experimental results:

1. The proposed $\ell_1$-MK-ELM and the radius-incorporated MK-ELM achieve statistically comparable or much better performance when compared with other state-of-the-art MKL algorithms. Moreover, the radius-incorporated MK-ELM can usually further improve the performance of $\ell_1$-MK-ELM. The improvement in classification performance is attributed to the manner in which it deals with multi-class tasks. Specifically, the MK-ELM is directly designed for solving multi-class problems while the OAA strategy is used in $\ell_1$-MK-SVM and $\ell_1$-MK-LSSVM. This allows MK-ELM to achieve better classification performance, as validated by the experimental results.
2. The proposed $\ell_1$-MK-ELM and the radius-incorporated MK-ELM are usually more computationally efficient than $\ell_1$-MK-SVM [19] and $\ell_1$-MK-LSSVM [30]. The computational efficiency of the proposed MK-ELM is due to the fact that only a matrix inverse operation is needed while $T$ ($T$ is the number of classes) constrained QP problems need to be solved at each iteration in $\ell_1$-MK-SVM and $\ell_1$-MK-LSSVM.
3. Considering the ratio of classification performance to computational cost, the proposed MK-ELM variants are clearly the best ones.

### 4.4. Comparison of non-sparse MKL algorithms

In this section, we compare the proposed $\ell_q$-MK-ELM with existing non-sparse MKL algorithms, including $\ell_q$-MK-SVM [20] and $\ell_q$-MK-FDA [24]. By following the setting in [24], the parameter $q$ which controls the sparsity of base kernel combination is taken from $\{32/31, 16/15, 8/7, 4/3, 2, 4, 8, 16\}$. The other settings follow the previous experiments.

#### 4.4.1. Results on protein fold prediction

Table 6 reports the experimental results of non-sparse MKL algorithms with different norms. From this table, we observe that:

- The proposed $\ell_q$-MK-ELM usually obtains comparable or better classification performance when compared with the others. Specifically, our proposed algorithm significantly outperforms

**Table 6**
Non-sparse MKL algorithms comparison with statistical test on the protein fold prediction data set. The four rows of each cell represent ACC/mAP, standard derivation, $p$-value and training time (in seconds), respectively. Boldface means no statistical difference from the best one ($p$-value $\geq 0.05$).

| Algorithms | $q=\frac{32}{31}$ | $q=\frac{16}{15}$ | $q=\frac{8}{7}$ | $q=\frac{4}{3}$ | $q=2$ | $q=4$ | $q=8$ | $q=16$ |
|---|---|---|---|---|---|---|---|---|
| ACC | | | | | | | | |
| $\ell_q$-MK-ELM (proposed) | 65.56 | **66.27** | **67.00** | **65.98** | **64.46** | **63.19** | **63.32** | **63.37** |
| | ± 1.66 | ± **1.20** | ± **1.13** | ± **1.07** | ± **0.61** | ± **0.99** | ± **0.95** | ± **0.99** |
| | 0.00 | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| | 1.85$e$+003 | 1.46$e$+003 | 1.05$e$+003 | 0.58$e$+003 | 0.35$e$+003 | 0.20$e$+003 | 0.15$e$+003 | 0.11$e$+003 |
| $\ell_q$-MK-SVM [20] | **65.74** | **65.98** | 66.05 | 64.78 | **64.31** | **63.08** | **63.03** | **62.90** |
| | ± **1.69** | ± **1.60** | ± 1.44 | ± 1.53 | ± **1.27** | ± **1.41** | ± **1.14** | ± **1.35** |
| | **1.00** | **0.27** | 0.00 | 0.01 | **0.72** | **0.85** | **0.39** | **0.18** |
| | 5.83$e$+003 | 5.12$e$+003 | 2.60$e$+003 | 0.98$e$+003 | 0.44$e$+003 | 0.22$e$+003 | 0.17$e$+003 | 0.13$e$+003 |
| $\ell_q$-MK-FDA [24] | 58.99 | 58.75 | 57.70 | 55.72 | 53.99 | 53.44 | 53.29 | 52.90 |
| | ± 1.69 | ± 1.42 | ± 1.30 | ± 1.31 | ± 1.93 | ± 1.67 | ± 1.59 | ± 1.68 |
| | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | 1.67$e$+003 | 0.76$e$+003 | 0.40$e$+003 | 0.15$e$+003 | 0.07$e$+003 | 0.07$e$+003 | 0.07$e$+003 | 0.06$e$+003 |
| mAP | | | | | | | | |
| $\ell_q$-MK-ELM (proposed) | **70.32** | **70.36** | **70.07** | **69.05** | **67.18** | **66.86** | **66.55** | **66.43** |
| | ± **1.17** | ± **1.49** | ± **0.93** | ± **0.89** | ± **1.14** | ± **1.08** | ± **0.91** | ± **0.95** |
| | **1.00** | **1.00** | **1.00** | **1.00** | **0.89** | **1.00** | **1.00** | **0.68** |
| $\ell_q$-MK-SVM [20] | 69.88 | 69.71 | 68.95 | 67.57 | 66.50 | 65.96 | 65.86 | 65.80 |
| | ± 1.09 | ± 1.04 | ± 0.83 | ± 1.17 | ± 1.09 | ± 1.06 | ± 1.02 | ± 1.04 |
| | 0.00 | 0.01 | 0.00 | 0.00 | 0.02 | 0.00 | 0.00 | 0.00 |
| $\ell_q$-MK-FDA [24] | 69.13 | 69.56 | 69.32 | 68.44 | **67.24** | **66.75** | **66.27** | **66.45** |
| | ± 1.07 | ± 1.03 | ± 0.86 | ± 1.11 | ± **1.29** | ± **1.06** | ± **1.00** | ± **0.96** |
| | 0.00 | 0.00 | 0.00 | 0.01 | **1.00** | **0.26** | **0.31** | **1.00** |

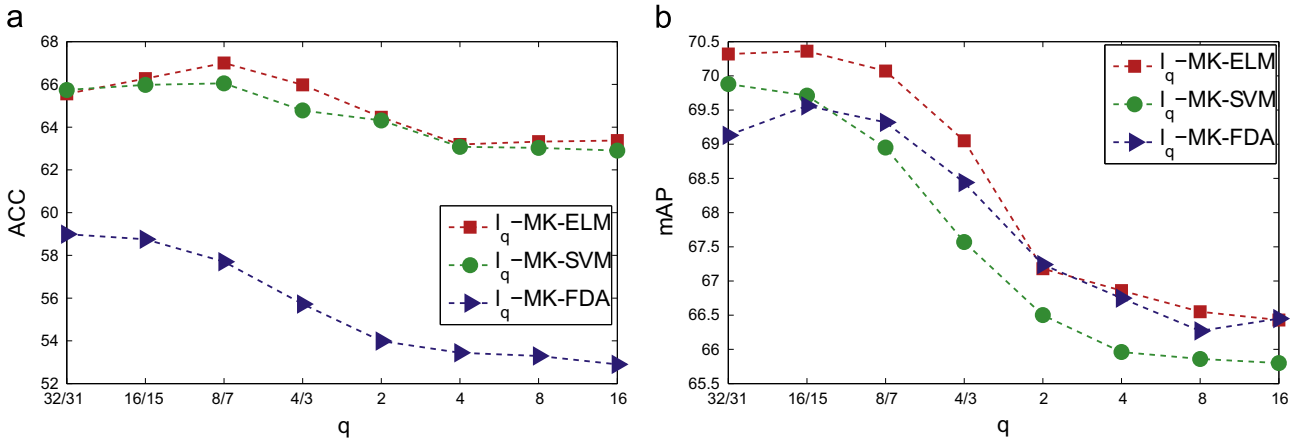


**Fig. 1.** Comparison classification performance of the proposed $\ell_q$-MK-ELM, $\ell_q$-MK-SVM [20] and $\ell_q$-MK-FDA [24] with the variation of $q$ on Protein Fold Prediction data set. (a) ACC. (b) mAP.

$\ell_q$-MK-FDA [24] in terms of both classification accuracy and mAP. Also, it shows significantly higher mAP and similar classification accuracy when compared with $\ell_q$-MK-SVM [20].

- The proposed $\ell_q$-MK-ELM is consistently more computationally efficient than $\ell_q$-MK-SVM [20] with the variation of $q$.

To better illustrate the experimental results, Fig. 1 plots the classification performance of the above MKL algorithms with the variation of $q$. As can be seen, the proposed $\ell_q$-MK-ELM is usually at the top, indicating its overall comparable or better classification performance. This validates the advantages of $\ell_q$-MK-ELM in handling multi-class classification problems.

#### 4.4.2. Results on Oxford Flower17

The results of non-sparse MKL algorithms on Oxford Flower17 are reported in Table 7. As can be observed, the proposed $\ell_q$-MK-ELM achieves comparable performance with $\ell_q$-MK-SVM [20] by incurring much less computational cost. In addition, the proposed algorithm shows significantly better classification performance when compared with $\ell_q$-MK-FDA [24] as $q$ increases. Similar results on the classification performance can also be found in Fig. 2.

#### 4.4.3. Results on Caltech101

We report the result on Caltech101 in Table 8. From this table, we observe that the proposed $\ell_q$-MK-ELM obtains significantly better mAP and comparable classification accuracy when compared with $\ell_q$-MK-SVM [20]. Furthermore, $\ell_q$-MK-ELM is statistically better than $\ell_q$-MK-FDA [24] in terms of both classification accuracy and mAP. In regard to the computational aspect, the training time required by $\ell_q$-MK-SVM [20] is approximately 40 times longer than that of $\ell_q$-MK-ELM, which evidences the computational efficiency of the proposed algorithm.

Fig. 3 shows three graphs corresponding to the classification performance of the three algorithms with the variation of $q$. From Fig. 3(b), we observe that our algorithm gains a significant improvement in terms of mAP over the other ones, while

**Table 7**
Non-sparse MKL algorithms comparison with statistical test on Oxford Flower17 data set. The four rows of each cell represent ACC/mAP, standard derivation, $p$-value and training time (in seconds), respectively. Boldface means no statistical difference from the best one ($p$-value $\geq 0.05$).

| Algorithms | $q=\frac{32}{31}$ | $q=\frac{16}{15}$ | $q=\frac{8}{7}$ | $q=\frac{4}{3}$ | $q=2$ | $q=4$ | $q=8$ | $q=16$ |
|---|---|---|---|---|---|---|---|---|
| ACC | | | | | | | | |
| $\ell_q$-MK-ELM (proposed) | **80.98** | **81.67** | **82.16** | **82.84** | **83.24** | **83.63** | **83.53** | **83.53** |
| | **±1.67** | **±2.38** | **±2.17** | **±1.72** | **±1.64** | **±1.87** | **±1.79** | **±1.79** |
| | **0.44** | **1.00** | **0.86** | **0.07** | **0.18** | **0.22** | **0.26** | **0.42** |
| | 0.48$e$+004 | 0.36$e$+004 | 2.12$e$+003 | 0.92$e$+003 | 0.61$e$+003 | 0.39$e$+003 | 0.23$e$+003 | 0.21$e$+003 |
| $\ell_q$-MK-SVM [20] | **81.47** | **81.57** | **82.25** | **83.43** | **83.82** | **83.92** | **83.92** | **83.82** |
| | **±0.78** | **±1.67** | **±1.45** | **±2.00** | **±1.79** | **±1.89** | **±1.89** | **±2.06** |
| | **1.00** | **0.83** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| | 1.25$e$+004 | 1.32$e$+004 | 7.69$e$+003 | 3.34$e$+003 | 1.55$e$+003 | 1.06$e$+003 | 0.67$e$+003 | 0.54$e$+003 |
| $\ell_q$-MK-FDA [24] | **78.92** | **78.14** | **78.73** | 79.80 | 80.59 | 80.59 | 80.59 | 80.59 |
| | **±0.95** | **±0.61** | **±0.74** | ±1.36 | ±1.28 | ±1.79 | 1.79 | ±1.79 |
| | **0.05** | **0.09** | **0.07** | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 |
| | 0.41$e$+004 | 0.38$e$+004 | 2.89$e$+003 | 0.45$e$+003 | 0.36$e$+003 | 0.41$e$+003 | 0.31$e$+003 | 0.28$e$+003 |
| mAP | | | | | | | | |
| $\ell_q$-MK-ELM (proposed) | 87.91 | 88.21 | **88.74** | **89.44** | **89.91** | **90.06** | **90.08** | **90.10** |
| | ±1.28 | ±1.39 | **±1.49** | **±1.58** | **±1.59** | **±1.52** | **±1.52** | **±1.52** |
| | 0.00 | 0.04 | **0.13** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| $\ell_q$-MK-SVM [20] | **88.40** | **88.50** | **88.84** | **89.36** | **89.73** | **89.92** | **89.97** | **90.00** |
| | **±1.22** | **±1.30** | **±1.46** | **±1.55** | **±1.50** | **±1.52** | **±1.56** | **±1.55** |
| | **1.00** | **1.00** | **1.00** | **0.35** | **0.12** | **0.15** | **0.15** | **0.23** |
| $\ell_q$-MK-FDA [24] | **87.73** | 87.10 | **88.03** | 88.69 | 89.71 | **90.02** | 90.06 | **90.08** |
| | **±1.40** | ±1.57 | **±1.01** | ±1.59 | ±1.59 | **±1.52** | ±1.52 | **±1.50** |
| | **0.26** | 0.01 | **0.16** | 0.00 | 0.00 | **0.05** | 0.04 | **0.29** |

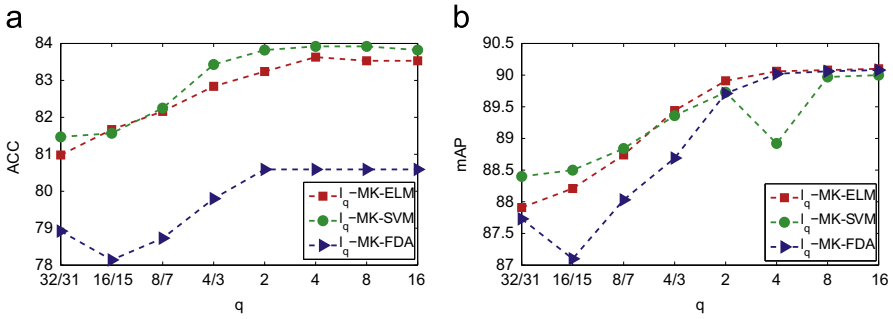


**Fig. 2.** Comparison classification performance of the proposed $\ell_q$-MK-ELM, $\ell_q$-MK-SVM [20] and $\ell_q$-MK-FDA [24] with the variation of $q$ on Oxford Flower17 data set. (a) ACC. (b) mAP.

**Table 8**
Non-sparse MKL algorithms comparison with statistical test on Caltech101 data set. The four rows of each cell represent ACC/mAP, standard derivation, $p$-value and training time (in seconds), respectively. Boldface means no statistical difference from the best one ($p$-value $\geq 0.05$).

| Algorithms | $q=\frac{32}{31}$ | $q=\frac{16}{15}$ | $q=\frac{8}{7}$ | $q=\frac{4}{3}$ | $q=2$ | $q=4$ | $q=8$ | $q=16$ |
|---|---|---|---|---|---|---|---|---|
| ACC | | | | | | | | |
| $\ell_q$-MK-ELM (proposed) | **64.05** | **63.89** | **63.75** | **63.63** | **63.81** | **63.69** | **63.78** | **63.80** |
| | **±1.64** | **±1.55** | **±1.51** | **±1.21** | **±1.07** | **±0.93** | **±0.93** | **±0.90** |
| | **0.64** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| | 0.18$e$+004 | 0.14$e$+004 | 0.11$e$+004 | 0.07$e$+004 | 0.43$e$+003 | 0.25$e$+003 | 0.19$e$+003 | 0.20$e$+003 |
| $\ell_q$-MK-SVM [20] | **64.12** | **63.74** | **63.54** | **63.47** | 63.35 | **63.34** | **63.27** | **63.27** |
| | **±1.85** | **±1.55** | **±1.27** | **±0.98** | ±0.88 | **±0.89** | **±0.92** | **0.91** |
| | **1.00** | **0.45** | **0.33** | **0.37** | 0.04 | **0.24** | **0.11** | **0.05** |
| | 5.91$e$+004 | 5.97$e$+004 | 3.59$e$+004 | 1.94$e$+004 | 9.44$e$+003 | 5.26$e$+003 | 3.98$e$+003 | 3.34$e$+003 |
| $\ell_q$-MK-FDA [24] | 56.45 | 56.30 | 56.11 | 57.56 | 60.42 | **61.82** | **61.82** | **61.82** |
| | ±1.77 | ±1.45 | ±2.32 | ±1.17 | ±1.69 | **±1.65** | **±1.65** | **1.65** |
| | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | **0.15** | **0.11** | **0.10** |
| | 0.03$e$+004 | 0.02$e$+004 | 0.02$e$+004 | 0.02$e$+004 | 0.22$e$+003 | 0.27$e$+003 | 0.24$e$+003 | 0.20$e$+003 |
| mAP | | | | | | | | |
| $\ell_q$-MK-ELM (proposed) | **65.29** | **64.97** | **64.74** | **64.73** | **64.83** | **64.89** | **64.89** | **64.88** |
| | **±1.74** | **±1.57** | **±1.40** | **±1.27** | **±1.20** | **±1.17** | **±1.17** | **±1.16** |
| | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| $\ell_q$-MK-SVM [20] | **64.59** | 64.14 | 63.70 | 63.55 | 63.39 | 63.34 | 63.30 | 63.28 |
| | **±2.01** | ±1.83 | ±1.50 | ±1.29 | ±1.27 | ±1.25 | ±1.24 | ±1.24 |
| | **0.05** | 0.03 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| $\ell_q$-MK-FDA [24] | 62.94 | 62.75 | 62.52 | 62.64 | 63.70 | 64.49 | 64.49 | 64.49 |
| | ±2.19 | ±1.65 | ±1.38 | ±1.16 | ±1.17 | ±1.13 | ±1.13 | ±1.13 |
| | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

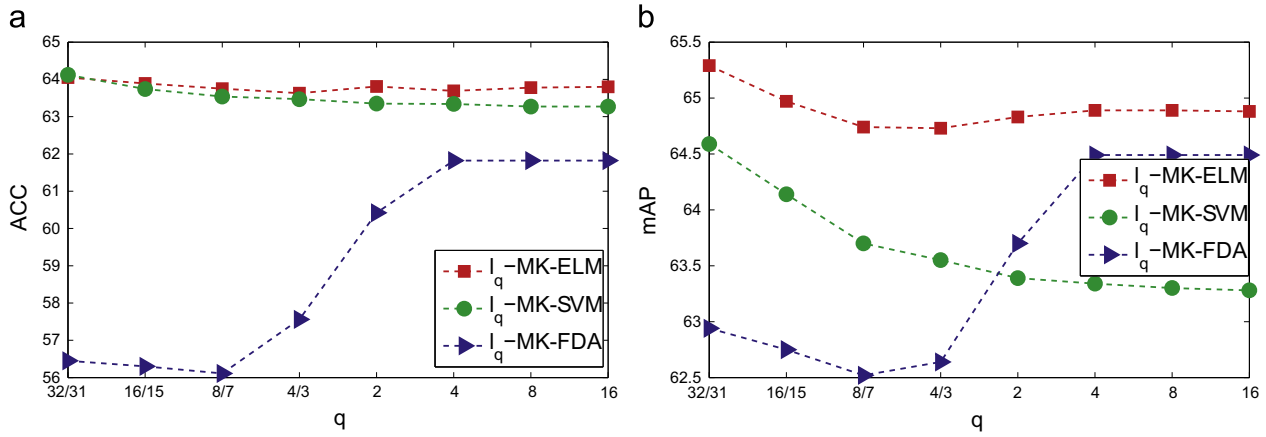


**Fig. 3.** Comparison classification performance of the proposed $\ell_q$-MK-ELM, $\ell_q$-MK-SVM [20] and $\ell_q$-MK-FDA [24] with the variation of $q$ on Caltech101 data set. (a) ACC. (b) mAP.

comparable or slightly better classification accuracy is achieved by our algorithm in Fig. 3(a).

Altogether, the better classification performance and computational efficiency of the proposed $\ell_q$-MK-ELM on protein fold predication, Oxford Flower17 and Caltech101 again demonstrates its advantage. In sum, the proposed MK-ELM algorithm is the overall best one when taking both classification performance and computational cost into account.

## 5. Conclusions

In this paper, we propose a multiple kernel extension for the extreme learning machine to allow it to better handle kernel tuning and heterogeneous data source integration. We first propose a sparse MK-ELM algorithm by imposing an $\ell_1$-norm constraint on the base kernel combination weights, and then extend it to the non-sparse case. After that, a radius-incorporated variant is proposed by integrating the radius information into ELM formulations. Three efficient algorithms are proposed to solve the resulting optimization problems. Comprehensive experiments are conducted to compare the proposed algorithms with existing state-of-the-art MKL algorithms in both sparse and non-sparse scenarios. As the experimental results indicate the proposed algorithms demonstrate comparable or better performance while requiring much less computational cost. In the future, we plan to improve the MK-ELM by integrating the correlation of base kernels into the formulations of existing MK-ELM [41].

## Acknowledgments

This work was supported by the National Basic Research Program of China (973) under Grant no. 2014CB340303, the National Natural Science Foundation of China (project nos. 61403405, 60970034, 61170287 and 61232016).

## Appendix A. The derivation of $\gamma$ for sparse MK-ELM

The Lagrangian function of sparse MK-ELM is

$$L_1(\tilde{\boldsymbol{\beta}},\boldsymbol{\xi},\boldsymbol{\gamma}) = \frac{1}{2}\sum_{p=1}^{m}\frac{\|\tilde{\boldsymbol{\beta}}_p\|_{\mathbf{F}}^2}{\gamma_p} + \frac{C}{2}\sum_{i=1}^{n}\|\boldsymbol{\xi}_{\cdot i}\|^2 - \sum_{t=1}^{T}\sum_{i=1}^{n}\alpha_{it}\left(\sum_{p=1}^{m}\tilde{\boldsymbol{\beta}}_p^{\top}\phi_p(\mathbf{x}_i) - y_{ti} + \xi_{ti}\right) + \tau\left(\sum_{p=1}^{m}\gamma_p - 1\right). \tag{A.1}$$

By taking the derivative of Eq. (A.1) with respect to $\gamma_p$ $(p=1,\cdots,m)$ and let it vanish, we obtained

$$-\frac{1}{2}\frac{\|\tilde{\boldsymbol{\beta}}_p\|_{\mathbf{F}}^2}{\gamma_p^2} + \tau = 0, \quad p = 1,\ldots,m. \tag{A.2}$$

By combining Eq. (A.2) and $\sum_{p=1}^{m}\gamma_p = 1$, we have

$$\gamma_p = \frac{\|\tilde{\boldsymbol{\beta}}_p\|_{\mathbf{F}}}{\sum_{p=1}^{m}\|\tilde{\boldsymbol{\beta}}_p\|_{\mathbf{F}}}, \quad p = 1,\cdots,m, \tag{A.3}$$

which completes the derivation.

## Appendix B. The derivation of $\gamma$ for non-sparse MK-ELM

The Lagrangian function of non-sparse MK-ELM is

$$L_2(\tilde{\boldsymbol{\beta}},\boldsymbol{\xi},\boldsymbol{\gamma}) = \frac{1}{2}\sum_{p=1}^{m}\frac{\|\tilde{\boldsymbol{\beta}}_p\|_{\mathbf{F}}^2}{\gamma_p} + \frac{C}{2}\sum_{i=1}^{n}\|\boldsymbol{\xi}_{\cdot i}\|^2 - \sum_{t=1}^{T}\sum_{i=1}^{n}\alpha_{it}\left(\sum_{p=1}^{m}\tilde{\boldsymbol{\beta}}_p^{\top}\phi_p(\mathbf{x}_i) - y_{ti} + \xi_{ti}\right) + \tau\left(\sum_{p=1}^{m}\gamma_p^q - 1\right). \tag{B.1}$$

By taking the derivative of Eq. (B.1) with respect to $\gamma_p$ $(p=1,\ldots,m)$ and let it vanish, we obtained

$$-\frac{1}{2}\frac{\|\tilde{\boldsymbol{\beta}}_p\|_{\mathbf{F}}^2}{\gamma_p^2} + q\tau\gamma_p^{q-1} = 0, \quad p = 1,\ldots,m. \tag{B.2}$$

By combining Eq. (B.2) and $\sum_{p=1}^{m}\gamma_p^q = 1$, we have

$$\gamma_p = \frac{\|\tilde{\boldsymbol{\beta}}_p\|_{\mathbf{F}}^{2/1+q}}{(\sum_{p=1}^{m}\|\tilde{\boldsymbol{\beta}}_p\|_{\mathbf{F}}^{2q/1+q})^{1/q}}, \quad p = 1,\ldots,m, \tag{B.3}$$

which completes the derivation.

## Appendix C. Proof of Theorem 1

The objective function of the radius-incorporated MK-ELM is presented in the following equation:

$$\min_{\boldsymbol{\gamma}}\min_{\tilde{\boldsymbol{\beta}},\boldsymbol{\xi}} \frac{1}{2}\left(\sum_{p=1}^{m}\gamma_p R_p^2\right)\sum_{p=1}^{m}\frac{\|\tilde{\boldsymbol{\beta}}_p\|_{\mathbf{F}}^2}{\gamma_p} + \frac{C}{2}\sum_{i=1}^{n}\|\boldsymbol{\xi}_{\cdot i}\|^2$$
$$\text{s.t.} \quad \sum_{p=1}^{m}\tilde{\boldsymbol{\beta}}_p^{\top}\phi_p(\mathbf{x}_i) = \mathbf{y}_i - \boldsymbol{\xi}_{\cdot i}, \quad i = 1,\ldots,n, \quad \gamma_p \geq 0,\ p = 1,\ldots,m. \tag{C.1}$$

Firstly, by defining $\boldsymbol{\beta}_p \triangleq \tilde{\boldsymbol{\beta}}_p/\sqrt{\gamma_p}$, $p=1,\ldots,m$, we reformulate Eq. (C.1) as

$$\min_{\boldsymbol{\gamma}} \mathcal{J}(\boldsymbol{\gamma}) \quad \text{s.t. } \gamma_p \geq 0, \quad p=1,\ldots,m, \tag{C.2}$$

where

$$\mathcal{J}(\boldsymbol{\gamma}) = \left\{ \min_{\boldsymbol{\beta},\boldsymbol{\xi}} \frac{1}{2}\left(\sum_{p=1}^{m}\gamma_p R_p^2\right)\sum_{p=1}^{m}\|\boldsymbol{\beta}_p\|_{\mathbf{F}}^2 + \frac{C}{2}\sum_{i=1}^{n}\|\boldsymbol{\xi}_{\cdot i}\|^2 \right.$$
$$\left. \text{s.t. } \boldsymbol{\beta}^{\top}\phi(\mathbf{x}_i;\boldsymbol{\gamma}) = \mathbf{y}_i - \boldsymbol{\xi}_{\cdot i}, \quad i=1,\ldots,n \right\}. \tag{C.3}$$

In order to prove Theorem 2, we firstly give the following Proposition 1. Its proof can be found in our previous work [27].

**Proposition 1.** *$\mathcal{J}(\mu\boldsymbol{\gamma}) = \mathcal{J}(\boldsymbol{\gamma})$, where $\mu$ is any positive scalar.*

**Theorem 2.** *Eq. (20) can be equivalently addressed by solving the following optimization problem:*

$$\min_{\boldsymbol{\gamma}} \min_{\tilde{\boldsymbol{\beta}},\boldsymbol{\xi}} \quad \frac{1}{2}\sum_{p=1}^{m}\frac{\|\tilde{\boldsymbol{\beta}}_p\|_{\mathbf{F}}^2}{\gamma_p} + \frac{C}{2}\sum_{i=1}^{n}\|\boldsymbol{\xi}_{\cdot i}\|^2$$
$$\text{s.t.} \quad \sum_{p=1}^{m}\tilde{\boldsymbol{\beta}}_p^{\top}\phi_p(\mathbf{x}_i) = \mathbf{y}_i - \boldsymbol{\xi}_{\cdot i}, \quad i=1,\ldots,n,$$
$$\sum_{p=1}^{m}\gamma_p R_p^2 = 1, \quad \gamma_p \geq 0, \ p=1,\ldots,m. \tag{C.4}$$

**Proof.** By defining $\widehat{\boldsymbol{\beta}_p} \triangleq \sqrt{(\sum_{p=1}^{m}\gamma_p R_p^2)}\boldsymbol{\beta}_p$, $(p=1,\ldots,m)$, Eq. (C.2) can be transformed to

$$\mathcal{J}(\boldsymbol{\gamma}) = \left\{ \min_{\widehat{\boldsymbol{\beta}},\boldsymbol{\xi}} \frac{1}{2}\sum_{p=1}^{m}\|\widehat{\boldsymbol{\beta}_p}\|_{\mathbf{F}}^2 + \frac{C}{2}\sum_{i=1}^{n}\|\boldsymbol{\xi}_{\cdot i}\|^2 \text{ s.t. } \widehat{\boldsymbol{\beta}}^{\top}\phi \right.$$
$$\left. \times\left(\mathbf{x}_i;\boldsymbol{\gamma}/\left(\sum_{p=1}^{m}\gamma_p R_p^2\right)\right) = \mathbf{y}_i - \boldsymbol{\xi}_{\cdot i}, \ i=1,\ldots,n. \right\} \tag{C.5}$$

Then, defining $\boldsymbol{\eta} \triangleq \boldsymbol{\gamma}/(\sum_{p=1}^{m}\gamma_p R_p^2)$, we have $\sum_{p=1}^{m}\eta_p R_p^2 = 1$. Also, by following Proposition 1, we can obtain $\mathcal{J}(\boldsymbol{\gamma}) = \mathcal{J}(\boldsymbol{\gamma}/(\sum_{p=1}^{m}\gamma_p R_p^2)) = \mathcal{J}(\boldsymbol{\eta})$. Hence, Eq. (C.5) can be written as

$$\mathcal{J}(\boldsymbol{\eta}) = \left\{ \min_{\widehat{\boldsymbol{\beta}},\boldsymbol{\xi}} \frac{1}{2}\sum_{p=1}^{m}\|\widehat{\boldsymbol{\beta}_p}\|_{\mathbf{F}}^2 + \frac{C}{2}\sum_{i=1}^{n}\|\boldsymbol{\xi}_{\cdot i}\|^2 \text{ s.t. } \widehat{\boldsymbol{\beta}}^{\top}\phi(\mathbf{x}_i;\boldsymbol{\eta}) = \mathbf{y}_i - \boldsymbol{\xi}_{\cdot i}, \quad i=1,\ldots,n \right\} \tag{C.6}$$

with constraints $\sum_{p=1}^{m}\eta_p R_p^2 = 1$, $\eta_p \geq 0$, $p=1,\ldots,m$, which is exactly the optimization problem in Eq. (C.4). This completes the proof. □

## Appendix D. The derivation of $\gamma$ for radius-incorporated MK-ELM

The Lagrangian function of radius-incorporated MK-ELM is as follows:

$$L_3(\tilde{\boldsymbol{\beta}},\boldsymbol{\xi},\boldsymbol{\gamma}) = \frac{1}{2}\sum_{p=1}^{m}\frac{\|\tilde{\boldsymbol{\beta}}_p\|_{\mathbf{F}}^2}{\gamma_p} + \frac{C}{2}\sum_{i=1}^{n}\|\boldsymbol{\xi}_{\cdot i}\|^2$$
$$- \sum_{t=1}^{T}\sum_{i=1}^{n}\alpha_{it}\left(\sum_{p=1}^{m}\tilde{\boldsymbol{\beta}}_p^{\top}\phi_p(\mathbf{x}_i) - y_{ti} + \xi_{ti}\right)$$
$$+ \tau\left(\sum_{p=1}^{m}\gamma_p R_p^2 - 1\right). \tag{D.1}$$

By taking the derivative of Eq. (D.1) with respect to $\gamma_p$ $(p=1,\ldots,m)$ and let it vanish, we obtained

$$-\frac{1}{2}\frac{\|\tilde{\boldsymbol{\beta}}_p\|_{\mathbf{F}}^2}{\gamma_p^2} + \tau R_p^2 = 0, \quad p=1,\ldots,m. \tag{D.2}$$

By combining Eq. (D.2) and $\sum_{p=1}^{m}\gamma_p R_p^2 = 1$, we have

$$\gamma_p = \frac{\|\tilde{\boldsymbol{\beta}}_p\|_{\mathbf{F}}}{R_p\sum_{p=1}^{m}R_p\|\tilde{\boldsymbol{\beta}}_p\|_{\mathbf{F}}}, \quad p=1,\ldots,m, \tag{D.3}$$

which completes the derivation.

## References


[1] G.-B. Huang, C.K. Siew, Extreme learning machine: RBF network case, in: ICARCV, 2004, pp. 1029–1036.
[2] G.-B. Huang, L. Chen, C.K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Netw. 17 (4) (2006) 879–892.
[3] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1–3) (2006) 489–501.
[4] G.-B. Huang, L. Chen, Convex incremental extreme learning machine, Neurocomputing 70 (16–18) (2007) 3056–3062.
[5] G.-B. Huang, L. Chen, Enhanced random search based incremental extreme learning machine, Neurocomputing 71 (16–18) (2008) 3460–3468.
[6] G.-B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Trans. Syst. Man Cybern. Part B 42 (2) (2012) 513–529.
[7] Q. Liu, Q. He, Z. Shi, Extreme support vector machine classifier, in: PAKDD, 2008, pp. 222–233.
[8] G. Feng, G.-B. Huang, Q. Lin, R.K.L. Gay, Error minimized extreme learning machine with growth of hidden nodes and incremental learning, IEEE Trans. Neural Netw. 20 (8) (2009) 1352–1357.
[9] R. Zhang, Y. Lan, G.-B. Huang, Z.-B. Xu, Universal approximation of extreme learning machine with adaptive growth of hidden nodes, IEEE Trans. Neural Netw. Learn. Syst. 23 (2) (2012) 365–371.
[10] G.-B. Huang, D. Wang, Advances in extreme learning machines (ELM2010), Neurocomputing 74 (16) (2010).
[11] G.-B. Huang, D. Wang, Y. Lan, Extreme learning machines: a survey, Int. J. Mach. Learn. Cybern. 2 (2011) 107–122.
[12] G.-B. Huang, D. Wang, Advances in extreme learning machines (ELM2011), Neurocomputing 102 (2011).
[13] W. Zong, G.-B. Huang, Y. Chen, Weighted extreme learning machine for imbalance learning, Neurocomputing 101 (2013) 229–242.
[14] Q. Yu, Y. Miche, E. Eirola, M. van Heeswijk, E. Séverin, A. Lendasse, Regularized extreme learning machine for regression with missing data, Neurocomputing 102 (2013) 45–51.
[15] Y. Chen, Z. Zhao, S. Wang, Z. Chen, Extreme learning machine-based device displacement free activity recognition model, Soft Comput. 16 (9) (2012) 1617–1625.
[16] B. Frénay, M. Verleysen, Using SVMs with randomised feature spaces: an extreme learning approach, in: ESANN, 2010.
[17] B. Frénay, M. Verleysen, Parameter-insensitive kernel in extreme learning for non-linear support vector regression, Neurocomputing 74 (16) (2011) 2526–2531.
[18] E. Parviainen, J. Riihimäki, Y. Miche, A. Lendasse, Interpreting extreme learning machine as an approximation to an infinite neural network, in: KDIR, 2010, pp. 65–73.
[19] A. Rakotomamonjy, F.R. Bach, S. Canu, Y. Grandvalet, SimpleMKL, J. Mach. Learn. Res. 9 (2008) 2491–2521.
[20] Z. Xu, R. Jin, H. Yang, I. King, M.R. Lyu, Simple and efficient multiple kernel learning by group Lasso, in: ICML, 2010, pp. 1175–1182.
[21] G.R.G. Lanckriet, N. Cristianini, P.L. Bartlett, L.E. Ghaoui, M.I. Jordan, Learning the kernel matrix with semidefinite programming, J. Mach. Learn. Res. 5 (2004) 27–72.
[22] Z. Xu, R. Jin, I. King, M.R. Lyu, An extended level method for efficient multiple kernel learning, in: NIPS, 2008, pp. 1825–1832.
[23] J. Ye, S. Ji, J. Chen, Multi-class discriminant kernel learning via convex programming, J. Mach. Learn. Res. 9 (2008) 719–758.
[24] F. Yan, J. Kittler, K. Mikolajczyk, A. Tahir, Non-sparse multiple kernel fisher discriminant analysis, J. Mach. Learn. Res. 13 (2012) 607–642 (ISSN 1532-4435).
[25] H. Do, A. Kalousis, A. Woznica, M. Hilario, Margin and radius based multiple kernel learning, in: Proceedings of the European Conference on Machine Learning, 2009, pp. 330–343.
[26] K. Gai, G. Chen, C. Zhang, Learning kernels with radiuses of minimum enclosing balls, in: Advances in Neural Information Processing Systems, vol. 23, 2010, pp. 649–657.
[27] X. Liu, L. Wang, J. Yin, E. Zhu, J. Zhang, An efficient approach to integrating radius information into multiple kernel learning, IEEE Trans. Cybern. 43 (2) (2013) 557–569.
[28] X. Wei, J. Löfberg, Y. Feng, Y. Li, Y. Li, Enclosing machine learning for class description, in: ISNN (1), 2007, pp. 424–433.
[29] D.M.J. Tax, R.P.W. Duin, Support vector data description, Mach. Learn. 54 (1) (2004) 45–66.
[30] S. Yu, T. Falck, A. Daemen, L.-C. Tranchevent, J.A.K. Suykens, B.D. Moor, Y. Moreau, L2-norm multiple kernel learning and its application to biomedical data fusion, BMC Bioinf. 11 (2010) 309.
[31] J. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Process. Lett. 9 (1999) 293–300.

[32] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[33] X. Liu, J. Yin, L. Wang, L. Liu, J. Liu, C. Hou, J. Zhang, An adaptive approach to learning optimal neighborhood kernels, IEEE Trans. Cybern. 43 (1) (2013) 371–384.
[34] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, Cambridge, 2004.
[35] H. Yang, Z. Xu, J. Ye, I. King, M.R. Lyu, Efficient sparse generalized multiple kernel learning, IEEE Trans. Neural Netw. 22 (3) (2011) 433–446.
[36] C. Davatzikos, P. Bhatt, L.M. Shaw, K.N. Batmanghelich, J.Q. Trojanowski, Prediction of MCI to AD conversion, via MRI, CSF biomarkers, and pattern classification, Neurobiol. Aging. 32 (12) (2011) 2322.e19–2322.e27. http://dx.doi.org/10.1016/j.neurobiolaging.2010.05.023.
[37] M. Gönen, E. Alpaydın, Multiple kernel learning algorithms, J. Mach. Learn. Res. 12 (July) (2011) 2211–2268.
[38] T. Damoulas, M.A. Girolami, Probabilistic multi-class multi-kernel learning: on protein fold recognition and remote homology detection, Bioinformatics 24 (10) (2008) 1264–1270.
[39] F. Yan, J. Kittler, K. Mikolajczyk, M.A. Tahir, Non-sparse multiple kernel fisher discriminant analysis, J. Mach. Learn. Res. 13 (2012) 607–642.
[40] M.-E. Nilsback, A. Zisserman, A visual vocabulary for flower classification, in: CVPR, 2006, pp. 1447–1454.
[41] C. Hinrichs, V. Singh, J. Peng, S. Johnson, Q-MKL: matrix-induced regularization in multi-kernel learning with applications to neuroimaging, in: NIPS, 2012, pp. 1430–1438.

**Xinwang Liu** received M.S. degree in Computer Science from the National University of Defense Technology, China in 2008. He is currently pursuing his Ph.D. degree in the same university. From October 2010, he spent 1 year in visiting the Engineering & Computer Science, the Australia National University, supported by the China Scholarship Council. From November 2011 to October 2012, he is a visiting student of the School of Computer Science and Software Engineering, University of Wollongong. His research interests focus on kernel learning and feature selection.

**Lei Wang** received the B.Eng. degree and the M.Eng. degree from Southeast University, China in 1996 and 1999, respectively, and the Ph.D. degree from Nanyang Technological University, Singapore in 2004. He worked as a research associate and research fellow at Nanyang Technological University from 2003 to 2005. After that, he joined the Australian National University and worked as a research fellow and then fellow from 2005 to 2011. In April 2011, he joined Faculty of Informatics University of Wollongong as a senior lecturer. He was awarded the Australian Post-doctoral Fellowship by Australian Research Council in 2007 and the Early Career Researcher Award by Australian Academy of Science in 2009, respectively. His research interests include machine learning, pattern recognition, and computer vision.

**Guang-bin Huang** received the B.Sc. degree in applied mathematics and M.Eng. degree in computer engineering from Northeastern University, PR China, in 1991 and 1994, respectively, and Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore in 1999. During undergraduate period, he also concurrently studied in Wireless Communication Department of Northeastern University, PR China. From June 1998 to May 2001, he worked as a research fellow in Singapore Institute of Manufacturing Technology (formerly known as Gintic Institute of Manufacturing Technology) where he has led/implemented several key industrial projects and also built up two R&D labs: Communication Information Technologies Lab and Mobile Communication Lab. He was the chief architect for several significant industrial projects including (Singapore Airlines) SATS Cargo Terminal 5 Information Tracking System. From May 2001, he has been working as an assistant professor and associate professor (tenured) in the School of Electrical and Electronic Engineering, Nanyang Technological University. He was a member of the Emergent Technologies Technical Committee of IEEE Computational Intelligence Society. He is a member of the Committee on Membership Development of IEEE Singapore Chapter. He serves as a Session Chair, Track Chair and Plenary Talk Chair for several international conferences. He is currently an associate editor of Neurocomputing, and IEEE Transactions on Systems, Man, and Cybernetics – Part B. He is a senior member of IEEE.

**Jian Zhang** received the B.Sc. degree from East Normal University, China in 1982; the M.Sc. degree in Computer Science from Flinders University, Australia in 1994; and the Ph.D. degree in Electrical Engineering from the University of New South Wales, Australia in 1999. From 1997 to 2003, he was with Visual Information Processing Lab, Motorola Labs in Sydney as a senior research engineer and later became a principal research engineer and foundation manager of Visual Communications Research Team. From 2004 to July 2011, he was a principal researcher, project leader in National ICT Australia (NICTA) and a conjoint associate professor with School of Computer Science and Engineering, University of New South Wales. He is currently an associate professor in Advanced Analytics Institute, Faculty of Engineering and Information Technology at University. His research interests include video analysis, indexing and search, and video coding and communication. He co-authored 90 paper publications, book chapters and 10 patents filed in US including five issued patents. He is an IEEE senior member, associated editors for IEEE Transactions on Circuits and Systems for Video Technology (T-CSVT) and EURASIP Journal on Image and Video Processing. He is a general co-chair to host the International Conference on Multimedia and Expo in Melbourne Australia 2012.

**Jianping Yin** received his M.S. degree and Ph.D. degree in Computer Science from the National University of Defense Technology, China, in 1986 and 1990, respectively. He is a full professor of computer science in the National University of Defense Technology. His research interests involve artificial intelligence, pattern recognition and algorithm design.