Neural Comput & Applic
DOI 10.1007/s00521-014-1709-7

Multiple-kernel-learning-based extreme learning machine for classification design

Xiaodong Li · Weijie Mao · Wei Jiang

Received: 17 September 2013 / Accepted: 9 September 2014

Abstract The extreme learning machine (ELM) is a new method for using single hidden layer feed-forward networks with a much simpler training method. While conventional kernel-based classifiers are based on a single kernel, in reality, it is often desirable to base classifiers on combinations of multiple kernels. In this paper, we propose the issue of multiple-kernel learning (MKL) for ELM by formulating it as a semi-infinite linear programming. We further extend this idea by integrating with techniques of MKL. The kernel function in this ELM formulation no longer needs to be fixed, but can be automatically learned as a combination of multiple kernels. Two formulations of multiple-kernel classifiers are proposed. The first one is based on a convex combination of the given base kernels, while the second one uses a convex combination of the so-called equivalent kernels. Empirically, the second formulation is particularly competitive. Experiments on a large number of both toy and real-world data sets (including high-magnification sampling rate image data set) show that the resultant classifier is fast and accurate and can also be easily trained by simply changing linear program.

X. Li · W. Mao (✉) · W. Jiang
State Key Laboratory of Industrial Control Technology Institute of Cyber-Systems and Control, Zhejiang University, Yuquan Campus, Hangzhou 310027, People's Republic of China
e-mail: wjmao@iipc.zju.edu.cn

X. Li
e-mail: hzxiaodong22@163.com

W. Jiang
e-mail: jiangwei@iipc.zju.edu.cn

Keywords Extreme learning machine · Multiple-kernel learning (MKL) · ELM kernel · Minimal norm of weights · QCQP · SILP · Multi-class ELM

1 Introduction

An optimal method to an automatic selection of optimal kernels is to learn a linear combination $K = \sum_{j=1}^{m} \mu_j K_j$ with mixing coefficients μ together with the model parameters. This framework, named multiple-kernel learning (MKL), was first introduced by [1] where two kinds of constraints on β and K have been considered, leading to either semi-definite programming or QCQP approaches, respectively. For appropriately designed sub-kernels K_j, the optimized combination coefficients can then be used to understand which features of the examples are of importance for discrimination: If one is able to obtain an accurate classification by a sparse weighting μ_j, then one can quite easily interpret the resulting decision function. Intuitively, sparseness of β makes sense when the expected number of meaningful kernels is small. Requiring that only a small number of features contribute to the final kernel implicitly, it assumes that most of the features to be selected are equally informative. In other words, sparseness is good when the kernels already contain a couple of good features that alone capture almost all of the characteristic traits of the problem. This also implies that features are highly redundant. It is now generally recognized as a powerful tool for various machine learning problems [2, 3].

Extreme learning machine (ELM) is proposed by Huang [4, 5]. ELM is a new type of single hidden layer feed-forward neural networks (SLFNs), and its core is a fixed hidden layer, which contains a large number of nonlinear nodes. The hidden layer bias of ELM is chosen randomly

Published online: 02 October 2014

beforehand, and only the output weights of ELM need to be calculated. The only factor which needs to be set by users is the size of ELM (the number of hidden nodes) [6]. Benefiting from the simple structure and effective training method, ELM has been successfully used in many practical tasks such as prediction, fault diagnosis, recognition, classification, signal processing, and so on.

Although ELM has made some achievements, but there is still space for improvement. Some scholars are engaged in optimizing the learning algorithm of ELM. Han et al. [7] encoded a priori information to improve the function approximation of ELM. Kim et al. [8] introduced a variable projection method to reduce the dimension of the parameter space. Zhu et al. [9] used a differential evolutionary algorithm to select the input weights of ELM.

Some other scholars dedicated to optimize the structure of ELM. Wang et al. [10] made a proper selection of the input weights and bias of ELM in order to improve the performance of ELM. Li et al. [11] proposed a structure-adjustable online ELM learning method, which can adjust the number of hidden layer RBF nodes. Huang et al. [12, 13] proposed an incremental structure ELM, which increases the hidden nodes gradually. Meanwhile, another incremental approach referred to as error-minimized extreme learning machine (EM-ELM) was proposed by Feng et al. [14]. All these incremental ELM start from a small size of ELM hidden layer and add random hidden node (nodes) to the hidden layer. During the growth of networks, the output weights are updated incrementally. On the other hand, an alternative method to optimize the structure of ELM is to train an ELM that is larger than necessary and then prune the unnecessarily nodes during learning. A pruned ELM (PELM) was proposed by Rong et al. [15, 16] for classification problem. Yoan et al. [17] proposed an optimally pruned extreme learning machine (OP-ELM) methodology. Besides, there are still other attempts to optimize the structure of ELM, such as CS-ELM [18] proposed by Lan et al., which used a subset model selection method. Zong et al. [29] put forward the weighted extreme learning machine for imbalance learning. The kernel trick applied to ELM was introduced in previous work [30].

Lanckriet et al. [1] pioneered the work of parameterized combinations of kernel learning in which the optimal kernel is obtained as a linear combination of pre-specified kernels. This procedure named as multiple-kernel learning (MKL) in the literature allows kernels to be chosen more automatic based on data and enhances the autonomy of machine learning process [27]. On the other hand, MKL is particularly valuable in application [32], i.e., through linear combination of kernel matrix learning, MKL offers the advantage of integrating heterogeneous data from multiple sources such as vectors, strings, trees, and graphs. Recently, MKL had been successfully applied to combine various heterogeneous data sources in practice [19]. Compared with the existing L_∞-norm MKL method, the L_2-norm MKL could lead to non-sparse solutions and more advantages when applied to biomedical problems. Ye et al. [20] showed that the multiple-kernel learning for LS-SVM can be formulated as a SDP problem, which has much less computational complexity compared with that of C-SVM. Sonnenburg et al. [22] applied a semi-infinite linear programming (SILP) strategy by reusing the SVM implementations for solving the subproblems inside the MKL optimization more efficiently, which made MKL applicable to large-scale data sets. Yang et al. [26] used the elastic net regularizer on the kernel combination coefficients as a constraint for MKL. Gu et al. [28] aimed to learn the map between the space of high-magnification sampling rate image patches and the space of blurred high-magnification sampling rate image patches based on the multi-kernel regression, which are the interpolation results generated from the corresponding low-resolution images. Liu et al. [33] designed sparse, non-sparse, and radius-incorporated MK-ELM algorithms.

We propose the issue of multiple-kernel learning for ELM by formulating it as a semi-infinite linear programming (SILP). In addition, the proposed algorithm optimizes the regularization parameter in a unified framework along with the kernels, which make the learning system more automatic. Empirical results on benchmark data sets prove that multiple-kernel learning for ELM (MKL-ELM) has good competitive performance compared with the traditional ELM algorithm.

2 Kernelized extreme learning machine

At first, we will briefly review the ELM proposed in [5, 6, 31]. The essence of ELM is that the hidden layer in ELM need not be tuned. The output function of ELM for generalized SLFNs is

$$f_L(x) = \sum_{i=1}^{L} \beta_i h_i(x_j) = \sum_{i=1}^{L} \beta_i h(w_i \cdot x_j + b_i) = h(x)\beta \quad j = 1, \ldots, N \tag{1}$$

where $w_i \in R^n$ is the weight vector connecting the input nodes and the ith hidden node, $b_i \in R$ is the bias of the ith hidden node, $\beta_i \in R$ is the weight connecting the ith hidden node and the output node, and $f_L(x) \in R$ is the output of the SLFNs. Where $w_i \cdot x_j$ denote the inner product of w_i and x_j, b_i are the learning parameters of hidden nodes, which are randomly chosen before learning.

If the standard SLFNs with at most $\tilde{N}$ hidden nodes can approximate theses N samples with zero error, it then means there exists β_i, w_i, and b_i such that

$$\sum_{i=1}^{L} \beta_i h(w_i \cdot x_j + b_i) = t_j, \quad j = 1, \ldots, N \tag{2}$$

Equation (2) can be written compactly as

$$\mathbf{H}\beta = \mathbf{T}. \tag{3}$$

Then, Eq. (3) become a linear system, and the smallest norm least squares solution of the linear system is

$$\beta = \mathbf{H}^{\dagger}\mathbf{T}, \tag{4}$$

where $\mathbf{H}^{\dagger}$ is the Moore–Penrose generalized inverse [4] of the hidden layer output matrix $\mathbf{H}$.

$$H = \begin{pmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{pmatrix} = \begin{pmatrix} h(w_1, b_1, x_1) & \cdots & h(w_L, b_L, x_1) \\ \vdots & \ddots & \vdots \\ h(w_1, b_N, x_1) & \cdots & h(w_L, b_L, x_N) \end{pmatrix}_{N \times L},$$

$\mathbf{T} = [t_1, \ldots, t_N]^T$, and $\beta = [\beta_1, \beta_2, \ldots, \beta_L]^T$.

$\mathbf{H}$ is called the hidden layer output matrix of the network [4, 5]; the ith column of $\mathbf{H}$ is the ith hidden node's output vector with respect to input $x_1, x_2, \ldots, x_N$; and the jth row of $\mathbf{H}$ is the output vector of the hidden layer with respect to input x_j.

It has been proven in [5, 6] that SLFNs with hidden nodes have the universal approximation capability; the hidden nodes can be randomly generalized in the beginning of learning. As introduced in [6], one of the methods to calculate Moore–Penrose generalized inverse of a matrix is the orthogonal projection method: $\mathbf{H}^{\dagger} = \mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}$.

According to the ridge regression theory [25], one can add a positive value to the diagonal of $\mathbf{H}\mathbf{H}^T$; the resultant solution is more stable and tends to have better generalization performance:

$$f(x) = h\beta = h(x)\mathbf{H}^T\left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{T}, \tag{5}$$

The feature mapping $h(x)$ is usually known to users in ELM. However, if a feature mapping $h(x)$ is unknown to users, a kernel matrix for ELM can be defined as follows [6]:

$$\Omega_{\mathrm{ELM}} = \mathbf{H}\mathbf{H}^T : \Omega_{\mathrm{ELM}_{i,j}} = h(x_i) \cdot h(x_j) = K(x_i, x_j). \tag{6}$$

Thus, the output function of kernel ELM classifier can be written compactly as:

$$f(x) = h(x)\mathbf{H}^T\left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{T} = \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_N) \end{bmatrix}^T \left(\frac{\mathbf{I}}{C} + \Omega_{\mathrm{ELM}}\right)^{-1}\mathbf{T}. \tag{7}$$

ELM is to minimize the training error as well as the norm of the output weights [6].

$$\text{Minimize}: \quad \|\mathbf{H}\beta - \mathbf{T}\|^2 \quad \text{and} \quad \|\beta\|. \tag{8}$$

In fact, according to (8), the classification problem for ELM with multi-output nodes can be formulated as:

$$\begin{aligned} \text{Minimize:} \quad & L_{P_{\mathrm{ELM}}} = \frac{1}{2}\|\beta\|^2 + C\frac{1}{2}\sum_{i=1}^{N}\|\xi_i\|^2 \\ \text{S.t.}: \quad & h(x)\beta = \mathbf{t}_i^T - \xi_i^T, \qquad i = 1, \ldots, N \end{aligned} \tag{9}$$

where $\xi_i = [\xi_{i,1}, \ldots, \xi_{i,m}]^T$ is the training error vector of the m output nodes with respect to the training sample x_i. Based on the KKT condition, to train ELM is equivalent to solving the following dual optimization problem:

$$\begin{aligned} \text{Minimize:} \quad & L_{D_{\mathrm{ELM}}} = \frac{1}{2}\|\beta\|^2 + C\frac{1}{2}\sum_{i=1}^{N}\|\xi_i\|^2 \\ & - \sum_{i=1}^{N}\sum_{j=1}^{m}\alpha_{i,j}(h(x_i)\beta_j - t_{i,j} + \xi_{i,j}). \end{aligned} \tag{10}$$

Algorithm 1: Given a training set $\{(x_i, t_i)\}_{i=1}^{N} \subset R^n \times R$, activation kernel function $g(\cdot)$, and the hidden node number L:

Step 1: Randomly assign input weight w_i and bias b_i, $i = 1, \ldots, L$.

Step 2: Calculate the hidden layer output matrix $\mathbf{H}$.

Step 3: Calculate the output weight β: $\beta = \mathbf{H}^{\dagger}\mathbf{T}$.

3 Multiple-kernel-learning algorithms for extreme learning machine

3.1 Minimum norm least squares (LS) solution of SLFNs

It is very interesting and surprising that unlike the most common understanding that all the parameters of SLFNs need to be adjusted, the input weights w_i and the hidden layer biases b_i are in fact not necessarily tuned, and the hidden layer output matrix $\mathbf{H}$ can actually remain unchanged once random values have been assigned to these parameters in the beginning of learning. For fixed input weights w_i and the hidden layer biases b_i, seen from Eq. (11), to train an SLFN is simply equivalent to finding a least squares solution $\hat{\beta}$ of the linear system $\mathrm{H}\beta = \mathbf{T}$:

$$\left\|\mathrm{H}(\hat{w}_1, \ldots, \hat{w}_{\tilde{N}}, \hat{b}_1, \ldots, \hat{b}_{\tilde{N}})\hat{\beta} - \mathrm{T}\right\| = \min_{w_i, b_i, \beta}\left\|\mathrm{H}(w_1, \ldots, w_{\tilde{N}}, b_1, \ldots, b_{\tilde{N}})\beta - T\right\| \tag{11}$$

$$\left\|\mathrm{H}(w_1,\ldots,w_{\tilde{N}},b_1,\ldots,b_{\tilde{N}})\hat{\beta}-\mathrm{T}\right\| = \min_{\beta}\left\|\mathrm{H}(w_1,\ldots,w_{\tilde{N}},b_1,\ldots,b_{\tilde{N}})\beta-\mathrm{T}\right\|. \quad (12)$$

If the number $\tilde{N}$ of hidden nodes is equal to the number N of distinct training samples $\tilde{N}=N$, matrix $\mathbf{H}$ is square and invertible when the input weight vectors w_i and the hidden biases b_i are randomly chosen, and SLFNs can approximate these training samples with zero error.

However, in most cases, the number of hidden nodes is much less than the number of distinct training samples $\tilde{N}\ll N$, H is a non-square matrix, and there may not exist $w_i,\ b_i,\ \beta_i\ (i=1,\ldots,\tilde{N})$ such that $\mathrm{H}\beta=\mathbf{T}$. The smallest norm least squares solution of the above linear system is

$$\hat{\beta}=\mathrm{H}^{\dagger}\mathrm{T}, \quad (13)$$

where $\mathrm{H}^{\dagger}$ is the Moore–Penrose generalized inverse of matrix $\mathbf{H}$ [21].

3.2 QCQP formulation

In the scenario of ELM, the duality gap of Eq. (9) and its dual program is zero since the Slater constraint qualification holds, and we get

$$\begin{aligned}&\min_{\omega,\xi}\ \max_{\alpha} L(\omega,\xi;\alpha)=\max_{\alpha}\ \min_{\omega,\xi} L(\omega,\xi;\alpha)\\ &\max\quad \alpha^T y-\frac{1}{2}\alpha^T K\alpha-\frac{1}{2C}\alpha^T\alpha \quad (14)\\ &\text{s.t.}\quad \sum\nolimits_{i=1}^{n}\alpha_i=0.\end{aligned}$$

The Lagrangian dual yields:

$$\min_{\lambda}\max_{\alpha}\quad \alpha^T y-\frac{1}{2}\alpha^T K\alpha-\frac{1}{2C}\alpha^T\alpha+\lambda\alpha^T 1_n,\quad \lambda\geq 0 \quad (15)$$

The Lagrangian function is given as

$$L(\lambda,\alpha)=\alpha^T y-\frac{1}{2}\alpha^T K\alpha-\frac{1}{2C}\alpha^T\alpha+\lambda\alpha^T\mathbf{1}_n,\ \ \lambda\geq 0 \quad (16)$$

To obtain the dual, the derivatives of the Lagrangian function with respect to the primal variables α have to vanish

$$\frac{\partial L(\lambda,\alpha)}{\partial\alpha}=0\ \ \rightarrow\ y-K\alpha-\frac{1}{C}\alpha+\lambda\mathbf{1}_n=0, \quad (17)$$

$$\begin{aligned}&\left(K+\frac{1}{C}I\right)\alpha=y+\lambda\mathbf{1}_n,\\ &\alpha=\left(K+\frac{1}{C}I\right)^{-1}(y+\lambda\mathbf{1}_n).\end{aligned} \quad (18)$$

According to (18), the Eq. (16) can be formulated as:

$$\begin{aligned}L(\lambda,\alpha)&=\alpha^T y-\frac{1}{2}\alpha^T(y+\lambda\mathbf{1}_n)+\lambda\alpha^T\mathbf{1}_n\\ &=\alpha^T y-\frac{1}{2}\alpha^T y-\frac{1}{2}\lambda\alpha^T\mathbf{1}_n+\lambda\alpha^T\mathbf{1}_n\\ &=\frac{1}{2}\alpha^T y+\frac{1}{2}\lambda\alpha^T\mathbf{1}_n\\ &=\frac{1}{2}\left(\left(K+\frac{1}{C}I\right)^{-1}(y+\lambda\mathbf{1}_n)\right)^T(y+\lambda\mathbf{1}_n)\\ &=\frac{1}{2}(y+\lambda\mathbf{1}_n)^T\left(K+\frac{1}{C}I\right)^{-1}(y+\lambda\mathbf{1}_n)\end{aligned} \quad (19)$$

We get its dual program:

$$\min_{\lambda}\frac{1}{2}(y+\lambda\mathbf{1}_n)^T\quad\left(K+\frac{1}{C}I\right)^{-1}(y+\lambda\mathbf{1}_n), \quad (20)$$

for the Slater constraint qualification holds. We consider the parameterization $K=\sum_{i=1}^{p}\mu_i K_i$ with additional affine constraint $\sum_{i=1}^{p}\mu_i=1$ and positive semi-definiteness constraint $K\geq 0$. Attaching these constraints into Eq. (20) and minimizing with respect to μ gives:

$$\begin{aligned}&\min_{\mu,\lambda}\quad \frac{1}{2}(y+\lambda\mathbf{1}_n)^T\left(\sum_{i=1}^{p}\mu_i K_i+\frac{1}{C}I\right)^{-1}(y+\lambda\mathbf{1}_n)\\ &s.t.\quad \sum_{i=1}^{p}\mu_i K_i\geq 0,\\ &\qquad \sum_{i=1}^{p}\mu_i=1,\end{aligned} \quad (21)$$

i.e.,

$$\begin{aligned}&\min_{\mu,\lambda,t}\quad t\\ &s.t.\quad \begin{pmatrix}\frac{1}{2}\sum_{i=1}^{p}\mu_i K_i+\frac{1}{2C}I & y+\lambda\mathbf{1}_n\\ (y+\lambda\mathbf{1}_n)^T & t\end{pmatrix}\geq 0,\\ &\qquad \sum_{i=1}^{p}\mu_i K_i\geq 0,\ \ \sum_{i=1}^{p}\mu_i=1.\end{aligned} \quad (22)$$

when the weights μ_i are constrained to be nonnegative and K_i are positive semi-definite, the constraint $\sum_{i=1}^{p}K_i\geq 0$ is satisfied naturally. In that case, we again consider the parameterization $K=\sum_{i=1}^{p}\mu_i K_i$ with constraint $\sum_{i=1}^{p}\mu_i=1$ and $\mu_i\geq 0$. Substituting this into Eq. (20) and minimizing with respect to μ give the following formulation:

$$\begin{aligned}
&\min_{\mu:\mu\geq 0,\mu^T\mathbf{1}_p=1}\ \max_{\alpha:\alpha^T\mathbf{1}_n=0}\left\{-\frac{1}{2}\sum_{i=1}^{p}\mu_i\alpha^T K_i\alpha+\alpha^T y-\frac{1}{2C}\alpha^T\alpha\right\}\\
&=\max_{\alpha:\alpha^T\mathbf{1}_n=0}\ \min_{\mu:\mu\geq 0,\mu^T\mathbf{1}_p=1}\left\{\alpha^T y-\frac{1}{2C}\alpha^T\alpha-\frac{1}{2}\sum_{i=1}^{p}\mu_i\alpha^T K_i\alpha\right\}\\
&=\max_{\alpha:\alpha^T\mathbf{1}_n=0}\left\{\alpha^T y-\frac{1}{2C}\alpha^T\alpha-\frac{1}{2}\max_{\mu:\mu\geq 0,\mu^T\mathbf{1}_p=1}\sum_{i=1}^{p}\mu_i\alpha^T K_i\alpha\right\}\\
&=\max_{\alpha:\alpha^T\mathbf{1}_n=0}\left\{\alpha^T y-\frac{1}{2C}\alpha^T\alpha-\frac{1}{2}\max_{p\geq i\geq 1}\alpha^T K_i\alpha\right\},
\end{aligned}\tag{23}$$

where $\mu_i^T\mathbf{1}^p=1$ means $\sum_{i=1}^{p}\mu_i=1$ (p is the number of kernels). Here, it should be noted that the order of the minimization and the maximization is interchanged in the first equation. This is right since the conditions that the objective is convex in μ and concave in α, the minimization problem is strictly feasible in μ, and the maximization problem is strictly feasible in α [1]. Equation (23) can be reformulated as the following QCQP

$$\begin{aligned}
\max_{\alpha,t}\quad & -\frac{1}{2}t+\alpha^T y-\frac{1}{2C}\alpha^T\alpha\\
\text{s.t.}\quad & t\geq\alpha^T K_i\alpha, \qquad i=1,\ldots,p,\\
& \alpha^T\mathbf{1}_n=0.
\end{aligned}\tag{24}$$

Such a QCQP problem can be solved efficiently with general-purpose optimization software packages, like MOSEK [23], which solve the primal and dual problems simultaneously using the interior point methods. The obtained dual variables can be used to fix the optimal kernel coefficients.

In some cases, the performance of ELM depends critically on the values of C. We show that the formulations (22) and (24) can be reformulated slightly, and this new formulation leads naturally to the estimation of the regularization parameter C in a joint framework. As can be seen from Eq. (22), the identity matrix appears in exactly the same form as kernel matrices; we can treat the regularization parameter as one of the coefficients for the kernel matrix and optimize them simultaneously. This leads to the following formulation:

$$\begin{aligned}
\min_{\mu,\lambda,t}\quad & t\\
\text{s.t.}\quad & \begin{pmatrix}\sum_{i=1}^{p}\mu_i K_i & y+\lambda\mathbf{1}_n\\ (y+\lambda\mathbf{1}_n)^T & t\end{pmatrix}\geq 0,\\
& \sum_{i=1}^{p}\mu_i K_i\geq 0,\ \sum_{i=0}^{p}\mu_i=1, \qquad \mu_0\geq 0,
\end{aligned}\tag{25}$$

where $\mu_0=\frac{1}{C}$ and $K_0=I$. To optimize the regularization parameter in Eq. (24), we modify Eq. (23) slightly:

$$\begin{aligned}
&\min_{\mu:\mu\geq 0,\mu^T\mathbf{1}_{p+1}=1}\ \max_{\alpha:\alpha^T\mathbf{1}_n=0}\left\{-\frac{1}{2}\sum_{i=0}^{p}\mu_i\alpha^T K_i\alpha+\frac{1}{2}\alpha^T y\right\}\\
&=\max_{\alpha:\alpha^T\mathbf{1}_n=0}\ \min_{\mu:\mu\geq 0,\mu^T\mathbf{1}_{p+1}=1}\left\{-\frac{1}{2}\sum_{i=0}^{p}\mu_i\alpha^T K_i\alpha+\frac{1}{2}\alpha^T y\right\}\\
&=\max_{\alpha:\alpha^T\mathbf{1}_n=0}\left\{\frac{1}{2}\alpha^T y-\frac{1}{2}\max_{\mu:\mu\geq 0,\mu^T\mathbf{1}_{p+1}=1}\sum_{i=0}^{p}\mu_i\alpha^T K_i\alpha\right\}\\
&=\max_{\alpha:\alpha^T\mathbf{1}_n=0}\left\{\frac{1}{2}\alpha^T y-\frac{1}{2}\max_{p\geq i\geq 0}\alpha^T K_i\alpha\right\},
\end{aligned}\tag{26}$$

where K_0 stands for unit matrix I, μ_0 denotes the reciprocal of regularization parameter. Substituting $\alpha^T K_i\alpha$ by t and moving it to the constraint, we get the following quadratically constraint linear program:

$$\begin{aligned}
\max\quad & \alpha^T y-t\\
\text{s.t.}\quad & t\geq\alpha^T K_i\alpha, \quad i=0,\ldots,p,\\
& \alpha^T\mathbf{1}_n=0.
\end{aligned}\tag{27}$$

We will show that the joint optimization of C works better in most cases in comparison with the approach of pre-specifying C.

3.3 SILP formulation

In this section, we show how these problems can be resolved by considering a novel dual formulation of the QCQP as a semi-infinite linear programming (SILP) problem. In the following formulation, K_j represents the jth kernel matrix in a set of $p+1$ kernels with the $(p+1)$th kernel identity matrix. The MKL-ELM is formulated as

$$\begin{aligned}
\max_{\theta,u}\quad & u\\
\text{s.t.}\quad & \theta_j\geq 0, && j=1,\ldots,p+1\\
& \sum_{j=1}^{p+1}\theta_j^2\leq 1,\\
& \frac{1}{2}\sum_{j=1}^{p+1}\theta_j f_j(\beta_q)-\frac{1}{2}\sum_{q=1}^{k}\beta_q^T Y_q^{-1}\mathbf{1}_q\geq u, && \forall\beta_q,\ q=1,\ldots,k\\
& f_j(\beta_q)=\sum_{q=1}^{k}\left(\frac{1}{2}\beta_q^T K_j\beta_q\right), && j=1,\ldots,p+1,\ q=1,\ldots,k
\end{aligned}\tag{28}$$

The MKL-ELM is presented in algorithm 2.

Algorithm 2: Obtain the initial guess $S^0 = 1$, $\beta^{(0)} = [\beta_1, \cdots, \beta_k]$, $\theta_j^{(0)} = \frac{1}{p+1}$, $\varepsilon = 10e-2$, $u^0 = -\infty$

While ($\Delta u > \varepsilon$)

Do

Step 1: Fix β and solve $\theta^{(l)}$ then obtain $u^{(l)}$

Step2: Compute kernel combination $\Omega^{(l)}$, where $\Omega^{(l)} = \sum_{j=1}^{p+1} \theta_j^{(l)} K_j$

Step3: Solve single ELM and obtain the optimal $\beta^{(l)}$

Step4: Compute $f_1(\beta^{(l)}), \cdots, f_{p+1}(\beta^{(l)})$, $s^{(l-1)} = \sum_{j=1}^{p+1} \theta_j^{(l-1)} f_j(\beta^{(l)})$

Step5: $\Delta u = \left| 1 - \frac{S^{(l-1)}}{u^{(l-1)}} \right|$, ($l$ is the pointer of the current repeat), Update

$(\theta^{(l+1)}, u^{(l+1)}) = \arg\max u$

$w.r.t\ \theta \in R_+^{p+1}, u \in R$ with $\sum_{j=1}^{p+1} \theta_j = 1$ and $\sum_{j=1}^{p+1} \theta_j S_j^r$ for $r = 1, \cdots, t$

End while

return (θ^*, β^*)

Step1 optimizes θ as a linear programming. Since the regularization coefficient is automatically estimated as θ_{p+1}, step 3 simplifies to a linear problem as

$$\begin{bmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & \Omega^{(l)} \end{bmatrix} \begin{bmatrix} 0 \\ \beta^{(l)} \end{bmatrix} = \begin{bmatrix} 0 \\ Y^{-1}\mathbf{1} \end{bmatrix} \tag{29}$$

4 Empirical results

In this section, we will perform classification experiments on such following data set: Banana, Breast Cancer, Titanic, Waveform, German, Image, Heart, Diabetes, Ringnorm, Thyroid, Twonorm, Flare Solar, and Splice (Table 1).

In the experiments, we will compare the following:

1. SK-ELM: single-kernel ELM
2. MK-ELM(MEB) [33]: the radius-incorporated multiple-kernel ELM

4.1 Multiple different kinds of kernels

Here, four sorts of different kernel functions, i.e., polynomial kernel function $k_1(x, y) = (1 + x^T y)^d$, Gaussian kernel function $k_2(x, y) = \exp\left(-\|x - y\|^2 \big/ \sigma^2\right)$, linear kernel function $k_3(x, y) = x^T y$, and Laplacian kernel function $= \exp(-\|x - y\|/p)$, are selected to construct the multiple kernels $k = \sum_{i=1}^{4} \mu_i k_i$, where the corresponding kernel parameters are specified as $d = 2$, $\sigma^2 = 20$, $p = 5$ before experiments. All kernel matrixes K_i are normalized through replacing $K_i(m, n)$ by $K_i(m, n) \big/ \sqrt{K_i(m, m) K_i(n, n)}$ to get unit diagonal matrix [1, 24]. Table 2 gives the results of the single-kernel ELM experiments on the Image, Ringnorm data set, etc. The criterion TRA (%) and TSA (%) represent the accuracy of the training set and of the testing set. As can be seen from Table 2, our proposed algorithm is a very potential tool. Most of the testing rates are high, basically attaining 90 % in the Laplacian kernel cases. About 88.66 % testing rate is obtained for the Ringnorm set in ELM cases, while for the image data set, such criterion is beyond 90.64 %. On the one hand, comparisons of Table 2 may suggest that the MKL-ELM has higher (at least the same) testing rate than the single kernel; on the other hand, the former is more time-consuming than the latter in parameter selection. Therefore, the MKL-ELM is proved again to be a more potential tool than the single-kernel one. Furthermore, most experimental results indicate that the proposed algorithm has higher accuracy, as well as the robust stability than the MK-ELM (MEB), SK-ELM, and ELM on the multi-class problems, which can be observed from the multiple-kernel experimental results in Table 2. Moreover, the proposed algorithm has less time-consuming than the MK-ELM (MEB).

Table 1 Data sets used in the experiments

Data set	Dimensionality	Training patterns	Test patterns
Banana	2	400	4,900
Breast cancer	9	200	77
Titanic	3	150	2,051
Waveform	21	400	4,600
German	20	700	300
Image	18	1,300	1,010
Heart	13	170	100
Diabetes	8	468	300
Ringnorm	20	400	7,000
Thyroid	5	140	75
Twonorm	20	400	7,000
Flare Solar	9	666	400
Splice	60	1,000	2,175

Table 3 reports the optimal kernel weight $\{\mu_i\}_{i=1}^4$ for every kernel function and the experimental results through MKL, i.e., the SILP formulation through MKL in the ELM-SILP. Summation of the average value of μ_1, μ_2, μ_3, μ_4 and μ_0 is not equal to 1. The optimal kernel weights μ_1, $\mu_2 = 0$, while μ_3, $\mu_4 \neq 0$ can be used to explain the cause why the MKL-ELM is likely to be better than the SK-ELM. Moreover, the TRA in the MKL-ELM can attain the maximum TRA in the SK-ELM. Higher TRA with MK will show better fitting capability of the MKL-ELM, while higher TSA will show better predication capacity. The above experimental results prove that the MKL-ELM has a lower upper bound of the expected risk and may give potential better data representation than the SK-ELM.

4.2 Fusing kernel experiments

In the section, the experiments are made through the fusing kernel learning (FKL) and exhibit the performance of the fusing kernel. In every feature set, five kinds of

Table 2 Experimental results of our method, MK-ELM(MEB), SK-ELM, and ELM: multi-class data sets

Data sets	Algorithm	Training times (s)	Testing	
			Rate (%)	Dev (%)
Banana	Our method	12.2237	89.62	8.22
	MK-ELM (MEB)	25.5569	89.02	5.16
	SK-ELM	0.1332	88.95	3.12
	ELM	0.0002	87.24	4.87
Breast cancer	Our method	30.4421	81.71	0.84
	MK-ELM (MEB)	70.2358	83.22	0.92
	SK-ELM	0.0095	85.78	1.01
	ELM	0.1291	88.23	2.07
Titanic	Our method	17.9782	82.03	9.22
	MK-ELM (MEB)	46.8891	81.15	8.21
	SK-ELM	0.1233	80.62	4.12
	ELM	0.0014	79.54	5.87
Waveform	Our method	320.5543	97.55	0.28
	MK-ELM (MEB)	700.3315	96.22	0.32
	SK-ELM	4.1734	92.83	0.23
	ELM	0.6513	90.72	1.21
German	Our method	250.6681	85.26	0.97
	MK-ELM (MEB)	360.5522	86.91	0.99
	SK-ELM	2.9965	88.92	1.02
	ELM	0.1355	92.70	1.63
Image	Our method	240.4451	96.81	0.28
	MK-ELM (MEB)	650.8841	95.17	0.28
	SK-ELM	3.0002	93.22	0.29
	ELM	0.0468	90.64	1.26
Heart	Our method	6.9982	93.27	4.27
	MK-ELM (MEB)	20.2256	91.26	5.34
	SK-ELM	0.0897	94.84	9.24
	ELM	0.0435	95.22	9.14
Diabetes	Our method	50.6342	93.95	0.62
	MK-ELM (MEB)	130.445	92.66	1.02
	SK-ELM	0.1235	91.63	0.92
	ELM	0.0062	88.71	2.17
Ringnorm	Our method	70.6523	96.52	0.26
	MK-ELM (MEB)	206.2241	94.61	0.25
	SK-ELM	0.1282	92.73	0.25
	ELM	0.05427	88.66	1.67
Thyroid	Our method	1.6123	97.82	8.31
	MK-ELM (MEB)	3.1245	96.22	6.53
	SK-ELM	0.0142	98.29	4.25
	ELM	0.0006	95.32	3.32
Twonorm	Our method	260.9782	99.6	0.28
	MK-ELM (MEB)	600.3722	97.21	0.29
	SK-ELM	4.173	92.57	0.29
	ELM	0.3725	88.75	1.53
Flare solar	Our method	21.0967	77.83	0.61
	MK-ELM (MEB)	60.1125	79.61	0.93
	SK-ELM	0.1128	82.65	0.89
	ELM	0.0092	83.26	2.56
Splice	Our method	852.27	88.71	0.09
	MK-ELM (MEB)	2,050.32	87.62	0.02
	SK-ELM	15.37	84.93	0.12
	ELM	0.7826	83.79	0.32

Gaussian kernel functions with bandwidth $\sigma^2 = [0.1 \quad 1 \quad 10 \quad 20 \quad 40]$ are used to form the fusing kernel $k = \sum_{i=1}^{2} \left(\sum_{j=1}^{5} \mu_{ij} k_{ij} \right)$. On the basis of the results mentioned above, it is clear that the fusing kernel not only can provide a good data representation of feature set, but also can distinguish which kernels are fit for the underlying problem. The perception into the reasoning made by the fusing kernel may be analyzed through the kernel weights reported in Table 4. The results of kernel weights, TRA, TSA, and CPU(s) are listed in Table 4. Obviously, there are very high TRA and TSA performance, which are obtained by the fusing kernel. Of course, it should be pointed out that the accuracy improvement on the fusing kernel model is at the cost of increasing the model complexity, which can be embodied from the increased experimental CPU time in Table 4.

Figure 1 shows the training time with varying number of samples for QCQP and SILP. For the max sample size, QCQP achieves 400 and the SILP approach achieves 1,300. For the training time, our approach costs the least time, three times faster than QCQP. In fact, our algorithm can solve various scale problems. We find that the termination of SILP is due to the problem of out of memory in Matlab,

Table 3 Experimental results of multiple different types of kernels

Data set	MK-ELM (Proposed)	μ_0^*	μ_1	μ_2	μ_3	μ_4	TRA (%)	TSA (%)
Banana	QCQP	1.000	0.008	0.000	0.088	0.000	87.5	87.8
	SILP	0.915	0.182	0.175	0.153	0.330	89.6	88.0
Breast cancer	QCQP	1.000	0.000	0.000	0.077	0.923	84.2	82.7
	SILP	0.986	0.010	0.013	0.079	0.000	82.3	81.7
Titanic	QCQP	1.000	0.199	0.223	0.000	0.578	82.3	82.0
	SILP	0.973	0.268	0.113	0.159	0.000	85.4	86.7
Waveform	QCQP	1.000	0.501	0.000	0.178	0.321	92.7	93.2
	SILP	0.898	0.229	0.110	0.325	0.071	94.8	97.5
German	QCQP	1.000	0.263	0.000	0.214	0.523	85.2	86.4
	SILP	0.828	0.213	0.131	0.219	0.123	83.6	85.2
Image	QCQP	1.000	0.157	0.000	0.401	0.442	97.7	96.2
	SILP	0.952	0.422	0.213	0.172	0.212	98.2	97.5
Heart	QCQP	1.000	0.226	0.612	0.162	0.000	89.4	90.2.
	SILP	0.982	0.134	0.213	0.279	0.058	92.7	93.9
Diabetes	QCQP	1.000	0.768	0.211	0.000	0.021	87.3	89.2
	SILP	0.957	0.182	0.175	0.153	0.330	91.6	92.2
Ringnorm	QCQP	1.000	0.218	0.113	0.248	0.390	95.2	94.1
	SILP	0.977	0.362	0.223	0.279	0.130	96.7	96.5
Thyroid	QCQP	1.000	0.223	0.332	0.247	0.198	92.7	93.5
	SILP	0.955	0.287	0.363	0.272	0.113	95.2	97.8
Twonorm	QCQP	1.000	0.872	0.012	0.024	0.192	97.8	98.2
	SILP	0.983	0.282	0.057	0.135	0.602	98.5	99.6
Flare Solar	QCQP	1.000	0.752	0.126	0.027	0.095	79.2	78.3
	SILP	0.920	0.419	0.015	0.193	0.228	79.3	77.8
Splice	QCQP	1.000	0.556	0.028	0.014	0.402	80.2	80.1
	SILP	0.962	0.322	0.256	0.323	0.226	86.4	88.7

* μ_0 is the reciprocal of the regularized parameter C

Table 4 Experimental results of fusing kernels

FKL	μ_0	Toy 1					Toy 2					TRA (%)	TSA (%)	CPU (s)
		μ_{11}	μ_{12}	μ_{13}	μ_{14}	μ_{15}	μ_{21}	μ_{22}	μ_{23}	μ_{24}	μ_{25}			
ELM-QCQP	0.373	0.385	0.015	0.028	0.089	0.000	0.000	0.051	0.059	0.000	0.000	98.6	97	7.622
ELM-SILP	0.399	0.399	0.397	0.228	0.165	0.139	0.399	0.392	0.222	0.148	0.118	99.8	99.2	60.262

and the termination of QCQP is due to the same problem in MOSEK.

The benchmark data set was constructed by 2,000 samples in Fig. 2. We used different kernel widths to construct the RBF kernel matrices and increase the number of kernel matrices from 2 to 200. The QCQP formulations had memory issues when the number of kernels was larger than 80.

In Fig. 3, the benchmark data were made up of two linear kernel matrices and 1,000 samples. The samples were equally and randomly divided into various numbers of classes. The number of classes gradually increased from 2 to 20.

5 Conclusions and future research

We investigate the issue of multiple-kernel learning based on ELM for classification in this paper. This problem is formulated as convex programs, and thus, globally optimal solutions are guaranteed. The final kernel functions for the multiple-kernel models are determined by black-box learning from the initially proposed kernel functions. Theoretically, the kernel functions used to construct the multiple-kernel models may be different for various problems, so the problem insight could impact on the selection of the kernel functions to improve the performance. In fact, some convex optimization problems are computationally

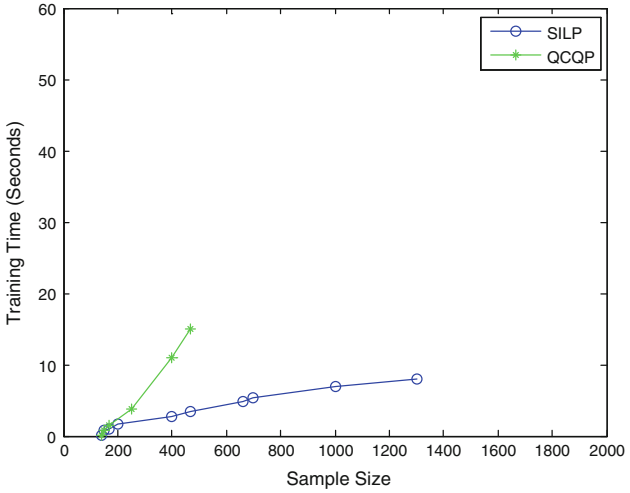

Fig. 1 Training time with varying number of samples for algorithms

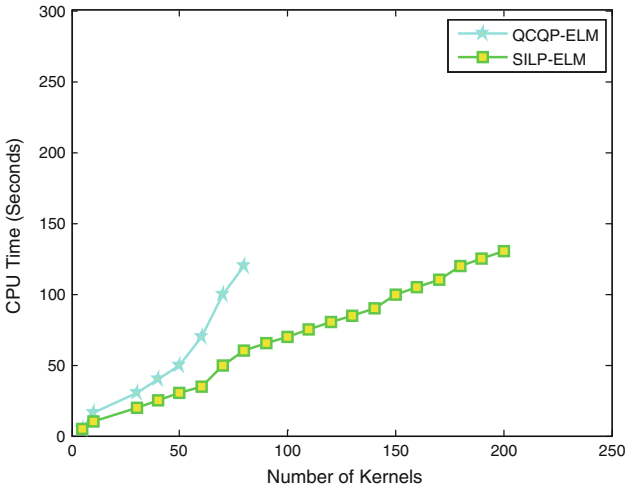

Fig. 2 Comparing the QCQP with the SILP formulations solve the MKL-ELM by different numbers of kernels

expensive, and the proposed algorithm is efficient to solve. Furthermore, we consider the problem of optimizing the kernel, thus approaching the desirable goal of automated model selection. To evaluate the proposed algorithms, we have conducted extensive experiments, which demonstrate the effectiveness.

The proposed MKL-ELM classifier is to learn the optimal combination of multiple large-scale data sets. Despite multiple-kernel ELM displaying some superiorities over the single-kernel ELM both theoretically and experimentally, there is much work worth investigating in the future, such as developing efficient algorithm to deal with large-scale training problem resulted from the increasing of the number of kernels.

From the practical point of view, our method can be easily applied in lots of applications, such as pattern recognition, time serial prediction, high-magnification sample image in painting, and bioinformatics.

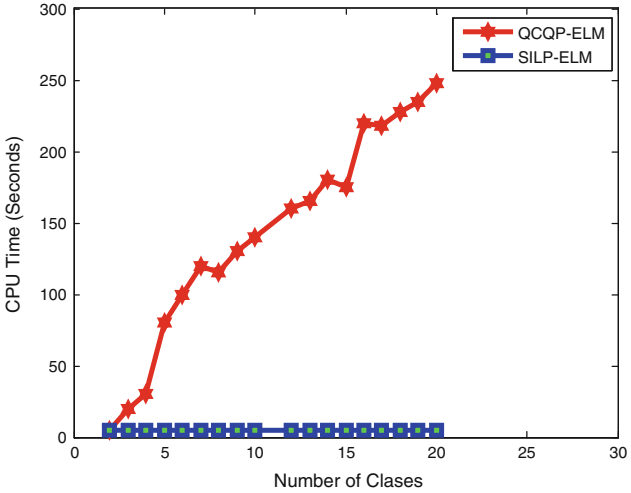

Fig. 3 Comparison of the QCQP and SILP formulations to solve the MKL-ELM data sets containing different numbers of classes

Acknowledgments We give warm thanks to Prof. Guangbin Huang, Prof. Zhihong Men, and the anonymous reviewers for helpful comments. This work was supported by the Zhejiang Provincial Natural Science Foundation of China (No. LR12F03002) and the National Natural Science Foundation of China (No. 61074045).

References

1. Lanckriet G, Cristianini N, Ghaoui LE, Bartlett P, Jordan MI (2004) Learning the kernel matrix with semi-definite programming. J Mach Learn Res 5:27–72
2. Schölkopf B, Smola AJ (2002) Learning with kernels. The MIT Press, Cambridge
3. Shawe-Taylor J, Cristianini N (2004) Kernel methods for pattern analysis. The Cambridge Univ Press, Cambridge
4. Huang GB, Chen L, Siew CK (2006) Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans Neural Netw 17:879–892
5. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70:489–501
6. Huang GB, Zhou H, Ding X, Zhang R (2012) Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern B Cybern 42:513–529
7. Han F, Huang DS (2006) Improved extreme learning machine for function approximation by encoding a priori information. Neurocomputing 69:2369–2373
8. Kim CT, Lee JJ (2008) Training two-layered feedforward networks with variable projection method. IEEE Trans Neural Netw 19:371–375
9. Zhu QY, Qin AK, Suganthan PN, Huang GB (2005) Evolutionary extreme learning machine. Pattern Recogn 38:1759–1763
10. Wang Y, Cao F, Yuan Y (2011) A study on effectiveness of extreme learning machine. Neurocomputing 74:2483–2490
11. Li GH, Liu M, Dong MY (2010) A new online learning algorithm for structure-adjustable extreme learning machine. Comput Math Appl 60:377–389

12. Huang GB, Chen L (2007) Convex incremental extreme learning machine. Neurocomputing 70:3056–3062
13. Huang GB, Li MB, Chen L, Siew CK (2008) Incremental extreme learning machine with fully complex hidden nodes. Neurocomputing 71:576–583
14. Feng G, Huang GB, Lin Q, Gay R (2009) Error minimized extreme learning machine with growth of hidden nodes and incremental learning. IEEE Trans Neural Netw 20:1352–1357
15. Rong HJ, Ong YS, Tan AH, Zhu Z (2008) A fast pruned-extreme learning machine for classification problem. Neurocomputing 72:359–366
16. Rong HJ, Huang GB, Sundararajan N, Saratchandran P (2009) Online sequential fuzzy extreme learning machine for function approximation and classification problems. IEEE Trans Syst Man Cybern B Cybern 39:1067–1072
17. Yoan M, Sorjamaa A, Bas P, Simula O, Jutten C, Lendasse A (2010) OP-ELM: optimally pruned extreme learning machine. IEEE Trans Neural Netw 21:158–162
18. Lan Y, Soh YC, Huang GB (2010) Constructive hidden nodes selection of extreme learning machine for regression. Neurocomputing 73:3191–3199
19. Yu S, Falck T, Daemen A et al (2010) L_2-norm multiple kernel learning and its application to biomedical data fusion. BMC Bioinformatics 11:1–53
20. Ye JP, Ji SW, Chen JH (2008) Multi-class discriminant kernel learning via convex programming. J Mach Learn Res 9:719–758
21. Serre D (2002) Matrices: Theory and Applications. Springer, New York
22. Sonnenburg S, Ratsch G, Schafer C, Scholkopf B (2006) Large scale multiple kernel learning. J Mach Learn Res 7:1531–1565
23. Andersen ED, Andersen AD (2000) The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In: Frenk H, Roos C, Terlaky T, Zhang S (eds) High performance optimization. Kluwer Academic Publishers, Norewell, USA, pp 197–232
24. Bach FR, Lanckriet GRG, Jordan MI (2004) Multiple kernel learning, conic duality, and the SMO algorithm. In: Proceedings of the 21st International Conference on Machine Learning (ICML). ACM, Banff, pp 6–13
25. Hoerl AE, Kennard RW (1970) Ridge regression: biased estimation for nonorthogonal problems. Technometrics 12:55–67
26. Yang H, Xu Z, Ye J, King I, Lyu MR (2011) Efficient sparse generalized multiple kernel learning. IEEE Trans Neural Netw 22:433–446
27. Gönen M, Alpaydın E (2011) Multiple kernel learning algorithms. J Mach Learn Res 12:2211–2268
28. Gu Y, Qu Y, Fang T, Li C, Wang H (2012) Image super-resolution based on multikernel regression. The 21st International Conference on in Pattern Recognition (ICPR)
29. Zong WW, Huang G-B, Chen Y (2013) Weighted extreme learning machine for imbalance learning. Neurocomputing 101: 229–242
30. Parviainen E, Riihimäki J, Miche Y, Lendasse A (2010) Interpreting extreme learning machine as an approximation to an infinite neural network. In: Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, pp 65–73
31. Huang GB, Wang D, Lan Y (2011) Extreme learning machines: a survey. Int J Mach Learn Cybernet 2:107–122
32. Hinrichs C, Singh V, Peng J, Johnson S (2012) Q-MKL: Matrix-induced regularization in multi-kernel learning with applications to neuroimaging. In: NIPS, pp 1430–1438
33. Liu X, Wang L, Huang G-B, Zhang J, Yin J (2013) Multiple kernel extreme learning machine. Neurocomputing. doi:10.1016/j.neucom.2013.09.072