Scalable Semi-Supervised Kernel Spectral Learning using Random Fourier Features

Siamak Mehrkanoon
KU Leuven, ESAT/STADIUS
B-3001 Leuven, Belgium
Email: siamak.mehrkanoon@esat.kuleuven.be
mehrkanoon2011@gmail.com

Johan A.K. Suykens
KU Leuven, ESAT/STADIUS
B-3001 Leuven, Belgium
Email: johan.suykens@esat.kuleuven.be

***Abstract*—We live in the era of big data with dataset sizes growing steadily over the past decades. In addition, obtaining expert labels for all the instances is time-consuming and in many cases may not even be possible. This necessitates the development of advanced semi-supervised models that can learn from both labeled and unlabeled data points and also scale at worst linearly with the number of examples. In the context of kernel based semi-supervised models, constructing the training kernel matrix for the large training dataset is expensive and memory inefficient. This paper investigates the scalability of the recently proposed multi-class semi-supervised kernel spectral clustering model (MSSKSC) by means of random Fourier features. The proposed model maps the input data into an explicit low-dimensional feature space. Thanks to the explicit feature maps, one can then solve the MSSKSC optimization formation in the primal, making the complexity of the method linear in number of training data points. The performance of the proposed model is compared with that of recently introduced reduced kernel techniques and Nyström based MSSKSC approaches. Experimental results demonstrate the scalability, efficiency and faster training computation times of the proposed model over conventional large scale semi-supervised models on large scale real-life datasets.**

I. Introduction

Learning methods are at the heart of many modern computer applications. A major obstacle in the successful use of completely supervised learning models is the need for sufficient expert-labeled instances. However, in many real-life applications, obtaining the labels of input data is cumbersome and expensive. Therefore in many cases one often encounters a large numbers of unlabeled data while the labeled data are rare.

Moreover, with the availability of abundant data, images and videos on the Internet, the size of the datasets has grown at a rapid rate [1]. Kernel based models have shown to be successful in many machine learning related tasks including classification, regression, clustering and semi-supervised learning among others. However, unfortunately, they scale poorly with the size of the training dataset due to the need for storing and computing the kernel matrix which is usually dense. The common solution is to approximate the kernel matrix using limited memory storage and in particular most of the kernel approximation methods such as Greedy basis selection techniques [2], [3], incomplete Cholesky decomposition [4], [3], [5], Nyström methods [6], [7], [8] aim at providing a low-rank approximation of the kernel matrix.

Besides the large scale data, in practice one also needs to address the issue of learning from a limited number of labeled instances and a huge amount of unlabeled data instances. Semi-Supervised Learning (SSL) is a framework in machine learning that aims at learning from both labeled and unlabeled data points [9]. Most of the developed semi-supervised approaches attempt to improve the performance by incorporating the information from either the unlabeled or labeled part. Graph based methods that assume that neighboring point pairs with a large weight edge are most likely within the same cluster. The Laplacian support vector machine (LapSVM) [10], is one of the graph based methods which provide a natural out-of-sample extension. Mehrkanoon *et al.* [11] proposed a multi-class semi-supervised algorithm (MSSKSC) where Kernel Spectral Clustering (KSC) is used as a core model. The available side-information (labels) is incorporated to the core model through a regularization term. In addition, the incremental MSSKSC for learning from a non-stationary environment is introduced in [12] where an adaptive mechanism is applied in order to update the learned model incrementally. Furthermore the extension of MSSKSC for classification of multi-label datasets with partially labeled instances are discussed in [13].

Many semi-supervised algorithms perform well on relatively small problems (see [14] and references therein), but they do not scale well when dealing with large scale data. Large scale semi-supervised modeling has not been considered in great detail in the literature. A family of semi-supervised linear support vector classifiers for large data sets is introduced in [15]. The authors in [16] introduced the prototype vector machine for large scale SSL. A large graph construction for scalable semi-supervised learning is proposed in [17]. Recently, the authors in [18], introduced two large scale semi-supervised algorithms, i.e. FS-MSSKSC and RD-MSSKSC, where the multi-class semi-supervised kernel spectral clustering (MSSKSC) serves as core model. FS-MSSKSC uses Nyström approximation to approximate the feature map and solves the optimization problem in the primal. Whereas RD-MSSKSC utilizes the reduced kernel technique and solves the optimization problem in the dual. In this paper we aim at yet making the recently proposed MSSKSC model introduced in [11] scalable for large scale problems using explicit random Fourier features.

978-1-5090-4240-1/16/$31.00 ©2016 IEEE

Comparison to the previously introduced large scale semi-supervised models is made to illustrate the efficiency and fast training computation times of the proposed model.

This paper is organized as follows. In Section II, a brief review of kernel spectral clustering (KSC) is given. Section III briefly reviews the multi-class semi-supervised kernel spectral clustering (MSSKSC) model and its existing large scale versions. Section IV, discusses the use of explicit random Fourier features in the MSSKSC formulation for large scale problems. Model selection aspects, simulation results as well as comparison with other large scale SSL models are discussed in Section V. The conclusion is given in Section VI.

II. KSC Model

The KSC method corresponds to a weighted kernel PCA formulation providing a natural extension to out-of-sample data i.e. the possibility to apply the trained clustering model to out-of-sample points. Given training data $\mathcal{D} = \{x_i\}_{i=1}^n$, $x_i \in \mathbb{R}^d$, the primal problem of kernel spectral clustering is formulated as follows [19]:

$$\begin{aligned} \min_{w_\ell, b_\ell, e_\ell} \quad & \frac{1}{2}\sum_{\ell=1}^{k-1} w^{(\ell)^T} w^{(\ell)} - \frac{1}{2n}\sum_{\ell=1}^{k-1} \gamma_\ell e^{(\ell)^T} V e^{(\ell)} \\ \text{subject to} \quad & e^{(\ell)} = \Phi w^{(\ell)} + b^{(\ell)} 1_n,\ \ell = 1, \ldots, k-1 \end{aligned} \tag{1}$$

where k is the number of desired clusters, $e^{(\ell)} = [e_1^\ell, \ldots, e_n^\ell]^T$ are the projected variables and $\ell = 1, \ldots, k-1$ indicates the number of score variables required to encode the k clusters. $\gamma_\ell \in \mathbb{R}^+$ are the regularization constants. Here

$$\Phi = [\varphi(x_1), \ldots, \varphi(x_n)]^T \in \mathbb{R}^{n \times h}$$

where $\varphi(\cdot) : \mathbb{R}^d \to \mathbb{R}^h$ is the feature map and h is the dimension of the feature space which can be infinite dimensional. A vector of all ones with size n is denoted by 1_n. $w^{(\ell)}$ is the model parameters vector in the primal. $V = \text{diag}(v_1, ..., v_n)$ with $v_i \in \mathbb{R}^+$ is a user defined weighting matrix.

Applying the Karush-Kuhn-Tucker (KKT) optimality conditions one can show that the solution in the dual can be obtained by solving an eigenvalue problem of the following form:

$$V P_v \Omega \alpha^{(\ell)} = \lambda \alpha^{(\ell)}, \tag{2}$$

where $\lambda = n/\gamma_\ell$, $\alpha^{(\ell)}$ are the Lagrange multipliers and P_v is the weighted centering matrix: $P_v = I_n - \frac{1}{1_n^T V 1_n} 1_n 1_n^T V$, where I_n is the $n \times n$ identity matrix and Ω is the kernel matrix with ij-th entry $\Omega_{ij} = K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$. In the ideal case of k well separated clusters, for a properly chosen kernel parameter, the matrix $V P_v \Omega$ has $k-1$ piecewise constant eigenvectors with eigenvalue 1.

The eigenvalue problem (2) is related to spectral clustering with random walk Laplacian. In this case, the clustering problem can be interpreted as finding a partition of the graph in such a way that the random walker remains most of the time in the same cluster with few jumps to other clusters, minimizing the probability of transitions between clusters. It is shown that if

$$V = D^{-1} = \text{diag}(\frac{1}{d_1}, ..., \frac{1}{d_n}),$$

where $d_i = \sum_{j=1}^n K(x_i, x_j)$ is the degree of the i-th data point, the dual problem is related to the random walk algorithm for spectral clustering.

From the KKT optimality conditions one can show that the score variables can be written as follows:

$$\begin{aligned} e^{(\ell)} &= \Phi w^{(\ell)} + b^{(\ell)} 1_n = \Phi \Phi^T \alpha^{(\ell)} + b^{(\ell)} 1_n \\ &= \Omega \alpha^{(\ell)} + b^{(\ell)} 1_n,\ \ell = 1, \ldots, k-1. \end{aligned}$$

The out-of-sample extensions to test points $\{x_i\}_{i=1}^{n_{\text{test}}}$ is done by an Error-Correcting Output Coding (ECOC) decoding scheme. First the cluster indicators are obtained by binarizing the score variables for test data points as follows:

$$\begin{aligned} q_{\text{test}}^\ell &= \text{sign}(e_{\text{test}}^\ell) = \text{sign}(\Phi_{\text{test}} w^{(\ell)} + b^{(\ell)} 1_{n_{\text{test}}}) \\ &= \text{sign}(\Omega_{\text{test}} \alpha^{(\ell)} + b^{(\ell)} 1_{n_{\text{test}}}), \end{aligned}$$

where $\Phi_{\text{test}} = [\varphi(x_1), \ldots, \varphi(x_{n_{\text{test}}})]^T$ and $\Omega_{\text{test}} = \Phi_{\text{test}} \Phi^T$. The decoding scheme consists of comparing the cluster indicators obtained in the test stage with the codebook (which is obtained in the training stage) and selecting the nearest codeword in terms of Hamming distance.

III. MSSKSC and its Large scale versions

A multi-class semi-supervised kernel spectral clustering (MSSKSC) is introduced in [11] where the information of the labeled instances are integrated to core KSC model via a regularization term. The MSSKSC model can operate in both semi-supervised classification and clustering modes by realizing a low dimensional embedding. An extension of the MSSKSC model to cope with large scale datasets are discussed recently in [18] where two methodologies one based on Nyström approximation of the feature map and the other one based on reduced kernel technique are introduced. Here we give a brief overview of the MSSKSC model and its large scale versions.

A. MSSKSC model

Consider training data points

$$\mathcal{D} = \{\underbrace{x_1, ..., x_{n_{UL}}}_{\substack{\text{Unlabeled} \\ (\mathcal{D}_U)}}, \underbrace{x_{n_{UL}+1}, .., x_n}_{\substack{\text{Labeled} \\ (\mathcal{D}_L)}}\}, \tag{3}$$

where $\{x_i\}_{i=1}^n \in \mathbb{R}^d$. The first n_{UL} data points do not have labels whereas the last $n_L = n - n_{UL}$ points have been labeled. Assume that there are Q classes, then the label indicator matrix $Y \in \mathbb{R}^{n_L \times Q}$ is defined as follows:

$$Y_{ij} = \begin{cases} +1 & \text{if the } i\text{th point belongs to the } j\text{th class,} \\ -1 & \text{otherwise.} \end{cases} \tag{4}$$

The formulation of multi-class semi-supervised KSC, in the primal, is given as follows [11]:

$$\begin{aligned} \min_{w^{(\ell)}, b^{(\ell)}, e^{(\ell)}} \quad & \frac{1}{2}\sum_{\ell=1}^{Q} w^{(\ell)^T} w^{(\ell)} - \frac{\gamma_1}{2}\sum_{\ell=1}^{Q} e^{(\ell)^T} V e^{(\ell)} + \\ & \frac{\gamma_2}{2}\sum_{\ell=1}^{Q} (e^{(\ell)} - c^{(\ell)})^T A (e^{(\ell)} - c^{(\ell)}) \\ \text{subject to} \quad & e^{(\ell)} = \Phi w^{(\ell)} + b^{(\ell)} 1_n,\ \ell = 1, \ldots, Q, \end{aligned} \tag{5}$$

where c^ℓ is the ℓ-th column of the matrix C defined as

$$C = [c^{(1)}, \ldots, c^{(Q)}]_{n \times Q} = \left[\frac{0_{n_{UL} \times Q}}{Y} \right]_{n \times Q}. \quad (6)$$

Here, $0_{n_{UL} \times Q}$ is a zero matrix of size $n_{UL} \times Q$ and Y is defined as previously. The matrix $\tilde{A}$ is defined as follows:

$$A = \left[\begin{array}{c|c} 0_{n_{UL} \times n_{UL}} & 0_{n_{UL} \times n_L} \\ \hline 0_{n_L \times n_{UL}} & I_{n_L \times n_L} \end{array} \right],$$

where $I_{n_L \times n_L}$ is the identity matrix of size $n_L \times n_L$. V is the inverse of the degree matrix defined as previously.

In optimization problem (5) the information of the labeled data is incorporated to the core KSC by means of a regularization term. The aim of this term is to minimize the squared distance between the projections of the labeled data and their corresponding labels. As illustrated in [11], given Q labels the approach is not restricted to finding just Q classes and instead is able to discover up to 2^Q hidden clusters. In addition, it uses low embedding dimension to reveal the existing number of clusters which is important when one deals with large number of clusters. Eliminating the primal variables $w^{(\ell)}, e^{(\ell)}$ and making use of Mercer's Theorem result in the following linear system of equations in the dual [11]:

$$\gamma_2 \left(I_n - \frac{R 1_n 1_n^T}{1_n^T R 1_n} \right) c^{(\ell)} = \alpha^{(\ell)} - R \left(I_n - \frac{1_n 1_n^T R}{1_n^T R 1_n} \right) \Omega \alpha^{(\ell)}, \quad (7)$$

where $R = \gamma_1 V - \gamma_2 A$.

One can notice that as in (5), in primal the feature map φ is not explicitly known, one uses the kernel trick to construct the full kernel matrix Ω. In addition, in the dual one has to solve a linear system of the same size as the number of training data points, i.e. n, (see Eq. (7)). Therefore for large scale data (n is large), it is not efficient to construct the full kernel matrix and solve a linear system of equations of size n. In order to overcome these practical issues, two possible approaches are proposed to in [18]. The first approach, Fixed-Size MSSKSC (FS-MSSKSC), is based on the Nyström approximation and the primal-dual formulation of the MSSKSC which is inspired on the fixed-size implementation of LSSVM formulation [20]. This is done by using a sparse approximation of the nonlinear mapping induced by the kernel matrix and solving the problem in the primal. The second approach, Reduced MSSKSC (RD-MSSKSC), is by means of the reduced kernel technique that solves the problem in the dual by reducing the dimensionality of the kernel matrix to a rectangular kernel. In what follows we give a brief overview of these two approaches.

B. Fixed-Size MSSKSC model (FS-MSSKSC)

The approach is based on the fact that one can obtain an explicit expression finite dimension for the feature map $\varphi(\cdot)$ by means of an eigenvalue decomposition of the kernel matrix Ω. As discussed in [18], in order to compute the approximated feature map, one applies the Nyström method to numerically solve the Fredholm integral equation of the first kind. This will lead to the following eigenvalue problem [8]:

$$\frac{1}{n} \sum_{k=1}^{n} K(x_k, x_j) u_{ik} = \lambda_i^{(s)} u_{ij} \quad (8)$$

where the eigenvalues and eigenfunctions of the continuous Fredholm integral equation is approximated by the sample eigenvalues $\lambda_i^{(s)}$ and eigenvectors u_i. Therefore, the i-th component of the n-dimensional feature map $\hat{\varphi} : \mathbb{R}^d \to \mathbb{R}^n$, for any point $x \in \mathbb{R}^d$, can be obtained as follows:

$$\hat{\varphi}_i(x) = \frac{1}{\sqrt{\lambda_i^{(s)}}} \sum_{k=1}^{n} u_{ki} K(x_k, x) \quad (9)$$

where $\lambda_i^{(s)}$ and u_i are eigenvalues and eigenvectors of the kernel matrix $\Omega_{n \times n}$. Furthermore, the k-th element of the i-th eigenvector is denoted by u_{ki}. In practice, when n is large, we work with a subsample (prototype vectors) of size $m \ll n$. There are several ways for which one can take to select the prototype vectors such as randomly, entropy based criterion [21], incomplete Cholesky factorization [5] and K-means clustering among others. The authors in [20], [18] used quadratic Rényi entropy for subset selection. In this case, the m-dimensional feature map $\hat{\varphi} : \mathbb{R}^d \to \mathbb{R}^m$ is approximated using $\hat{\varphi}(x) = [\hat{\varphi}_1(x), \ldots, \hat{\varphi}_m(x)]^T$, where

$$\hat{\varphi}_i(x) = \frac{1}{\sqrt{\lambda_i^{(s)}}} \sum_{k=1}^{m} u_{ki} K(x_k, x), i = 1, \ldots, m \quad (10)$$

where $\lambda_i^{(s)}$ and u_i are now eigenvalues and eigenvectors of the constructed kernel matrix $\Omega_{m \times m}$ using the selected prototype vectors. Given the m-dimensional approximation to the feature map, i.e. $\hat{\Phi} = [\hat{\varphi}(x_1), \ldots, \hat{\varphi}(x_n)]^T \in \mathbb{R}^{n \times m}$, the optimization problem (5) can be rewritten as an unconstrained optimization problem. Therefore, one can seek the solution by solving the optimization problem in the primal [20]. This leads to solving the following linear system of equations [18]:

$$\left[\begin{array}{c} w^{(\ell)} \\ b^{(\ell)} \end{array} \right] = \left(\Phi_e^T R \Phi_e + I_{(m+1)} \right)^{-1} \gamma_2 \Phi_e^T c^{(\ell)}, \ell = 1, \ldots, Q, \quad (11)$$

where $R = \gamma_2 A - \gamma_1 V$ is a diagonal matrix, $\Phi_e^T = \left[\begin{array}{c} \hat{\Phi}^T \\ 1_n^T \end{array} \right]_{(m+1) \times n}$ and $I_{(m+1)}$ is the identity matrix of size $(m+1) \times (m+1)$. One should note that the solution vector $w^{(\ell)}$ obtained by FS-MSSKSC has the same dimension as the number of prototype vectors. The score variables evaluated at the test set $\mathcal{D}^{\text{test}} = \{x_i\}_{i=1}^{n_{\text{test}}}$ become [18]:

$$e_{\text{test}}^{(\ell)} = \hat{\Phi}_{\text{test}} w^{(\ell)} + b^{(\ell)} 1_{n_{\text{test}}} \ \ell = 1, \ldots, Q, \quad (12)$$

where $\hat{\Phi}_{\text{test}} = [\hat{\varphi}(x_1), \ldots, \hat{\varphi}(x_{n_{\text{test}}})]^T \in \mathbb{R}^{n_{\text{test}} \times m}$. The decoding scheme consists of comparing the binarized score variables for test data points with a codebook obtained using the training labeled data and selecting the nearest codeword in terms of Hamming distance.

C. Reduced MSSKSC model (RD-MSSKSC)

The practical difficulty of solving the MSSKSC formulation (7) in the dual results into the huge kernel matrix which cannot be stored into memory. A reduced kernel technique is used in [18] to solve the optimization problem (5) in the dual with a rectangular kernel matrix. In reduced MSSKSC model (RD-MSSKSC), proposed in [18], as opposed to FS-MSSKSC, one does not need to apply the eigen-decomposition of the kernel matrix associated with the prototype vectors to obtain the explicit feature map. The approach overcomes the difficulty of storing the large scale kernel matrix by reducing the $n \times n$ dimensionality of the kernel matrix Ω to a much smaller dimensionality of a rectangular kernel matrix $\bar{\Omega} \in \mathbb{R}^{n \times \bar{n}}$ with $\bar{\Omega}_{ij} = K(x_i, x_j)$ and $x_i \in X$ and $x_j \in \bar{X}$. Here $\bar{X}$ is a $(\bar{n} \times d)$ random submatrix of the matrix of training data points X. In [18] the subset is selected by means of a Rényi entropy based criterion [22]) and by using the Sherman-Morrison-Woodbury formula [4], the solution in the dual is obtained as follows [18]:

$$\beta^{(\ell)} = \left[I_n - R\bar{G} \left(I_{\bar{n}+1} + \bar{G}^T R \bar{G} \right)^{-1} \bar{G}^T \right] \gamma_2 c^{(\ell)}, \ell = 1, \ldots, Q, \tag{13}$$

where R is defined as previously, $\bar{G} = [\bar{\Omega}, 1_n] \in \mathbb{R}^{n \times (\bar{n}+1)}$ and I_n is the identity matrix. The expression (13) involves the inversion of a small matrix of order $(\bar{n}+1) \times (\bar{n}+1)$. One may notice that the solution vector $\beta^{(\ell)}$ obtained by RD-MSSKSC has the same dimension as the number of training points. The bias term $b^{(\ell)}$ for $\ell = 1, \ldots, Q$ can be computed based on the one of the KKT optimality conditions as follows [18]:

$$b^{(\ell)} = 1_n^T \beta^{(\ell)}, \ \ell = 1, \ldots, Q.$$

After obtaining the $\beta^{(\ell)}$ and $b^{(\ell)}$, one can compute the score variables of the test set $X^{\text{test}} = \{x_i\}_{i=1}^{n_{\text{test}}}$ as follows [18]:

$$\begin{aligned} e_{\text{test}}^{(\ell)} &= \bar{\Omega}^{\text{test}} \, \alpha^{(\ell)} + b^{(\ell)} 1_{n_{\text{test}}} \\ &= \left[\bar{\Omega}^{\text{test}} \ \bar{\Omega}^T \right] \beta^{(\ell)} + b^{(\ell)} 1_{n_{\text{test}}}, \ \ell = 1, \ldots, Q, \end{aligned} \tag{14}$$

where $\bar{\Omega}_{ij}^{\text{test}} = K(x_i, x_j)$ with $x_i \in X^{\text{test}}$ and $x_j \in \bar{X}$. The decoding scheme consists of comparing the binarized score variables for test data points with a codebook obtained using the training labeled data and selecting the nearest codeword in terms of Hamming distance.

IV. MSSKSC with Random Fourier Features

The fundamental building block of the theory of kernel based approaches is the kernel function, which computes the similarity of multidimensional data points. Usually these approaches use an implicit feature mapping in the primal level, and a kernel trick in the dual to compute the full kernel matrix. However, for large scale problems it is not computationally efficient to build the entire kernel matrix and therefore many efforts have been done to deliver large-scale versions of kernel machines which some of them are discussed in section III.

This section explores an alternative pathway, i.e. randomization. An alternative to reduced rank approximations has been recently introduced in the field of kernel methods by exploiting the classical Bochner's theorem in harmonic analysis [23]. The Bochner's theorem states that a continuous kernel $K(x, y) = K(x - y)$ on $\mathbb{R}^d$ is positive definite if and only if K is the Fourier transform of a non-negative measure. If a shift-invariant kernel k is properly scaled, its Fourier transform $p(\xi)$ is a proper probability distribution. This property is used to approximate kernel functions with linear projections on D random features as follows [23]:

$$K(x - y) = \int_{\mathbb{R}^d} p(\xi) e^{j\xi^T (x-y)} d\xi = \mathbb{E}_\xi [z_\xi(x) z_\xi(y)^*], \tag{15}$$

where $z_\xi(x) = e^{j\xi^T x}$. Here $z_\xi(x) z_\xi(y)^*$ is an unbiased estimate of $K(x, y)$ when ξ is drawn from $p(\xi)$ (see [23]). To obtain a real-valued random feature for K, one can replace the $z_\xi(x)$ by the mapping $z_\xi(x) = [\cos(\xi^T x), \sin(\xi^T x)]$ which also satisfies the condition $\mathbb{E}_\xi[z_\xi(x) z_\xi(y)^*]$. The random Fourier feature $z(x)$, for the sample x, is then defined as $z(x) = \frac{1}{\sqrt{D}} [z_{\xi_1}(x), \ldots, z_{\xi_D}(x)]^T \in \mathbb{R}^{2D}$ (see [23]). Here $\frac{1}{\sqrt{D}}$ is used as normalization factor to reduce the variance of the estimate and $\xi_1, \ldots, \xi_D \in \mathbb{R}^d$ are sampled from $p(\xi)$. For a Gaussian kernel, they are drawn from a Normal distribution $\mathcal{N}(0, I_d/\sigma^2)$. One can now construct the explicit feature map for the entire training dataset using the finite dimensional random Fourier features as follows:

$$\Phi_{\text{RFF}} = [z(x_1), \ldots, z(x_n)]^T \in \mathbb{R}^{n \times 2D}. \tag{16}$$

Given Φ_{RFF}, one can rewrite the optimization problem (5) as an unconstrained optimization problem and solve it in primal:

$$\begin{aligned} \min_{w^{(\ell)}, b^{(\ell)}} \mathcal{J}(w^{(\ell)}, b^{(\ell)}) = \frac{1}{2} \sum_{\ell=1}^{Q} {w^{(\ell)}}^T w^{(\ell)} - \\ \frac{\gamma_1}{2} \sum_{\ell=1}^{Q} (\Phi_{\text{RFF}} w^{(\ell)} + b^{(\ell)} 1_n)^T V (\Phi_{\text{RFF}} w^{(\ell)} + b^{(\ell)} 1_n) + \\ \frac{\gamma_2}{2} \sum_{\ell=1}^{Q} (c^{(\ell)} - \Phi_{\text{RFF}} w^{(\ell)} - b^{(\ell)} 1_n)^T A (c^{(\ell)} - \Phi_{\text{RFF}} w^{(\ell)} - b^{(\ell)} 1_n) \end{aligned} \tag{17}$$

where the matrix C is defined as previously in (6). Taking the partial derivatives of the cost function $\mathcal{J}$ with respect to the primal variables $w^{(\ell)}$ and $b^{(\ell)}$ yields:

$$\begin{cases} \frac{\partial \mathcal{J}}{\partial w^{(\ell)}} = 0 \rightarrow (I + \Phi_{\text{RFF}}^T R \Phi_{\text{RFF}}) w^{(\ell)} + \Phi_{\text{RFF}}^T R 1_n b^{(\ell)} = \\ \qquad\qquad \gamma_2 \Phi_{\text{RFF}}^T c^{(\ell)}, \ \ell = 1, \ldots, Q, \\ \frac{\partial \mathcal{J}}{\partial b^{(\ell)}} = 0 \rightarrow 1_n^T R \Phi_{\text{RFF}} w^{(\ell)} + (1_n^T R 1_n) b^{(\ell)} = \\ \qquad\qquad \gamma_2 1_n^T c^{(\ell)}, \ \ell = 1, \ldots, Q, \end{cases} \tag{18}$$

which then by using some algebraic manipulations can be rewritten as a linear system of equations in terms of the primal variables as follows:

$$\begin{bmatrix} w^{(\ell)} \\ b^{(\ell)} \end{bmatrix} = \left(\tilde{\Phi}_{\text{RFF}}^T R \tilde{\Phi}_{\text{RFF}} + I_{((2D)+1)} \right)^{-1} \gamma_2 \tilde{\Phi}_{\text{RFF}}^T c^{(\ell)}, \tag{19}$$

for $\ell = 1, \ldots, Q$. Here $R = \gamma_2 A - \gamma_1 V$ is a diagonal matrix, $\tilde{\Phi}_{\text{RFF}}^T = \left[\begin{array}{cc} \Phi_{\text{RFF}}, & 1_n \end{array} \right]^T \in \mathbb{R}^{(2D+1)\times n}$ and $I_{(2D+1)}$ is the identity matrix of size $(2D+1) \times (2D+1)$.

The codebook $\mathcal{CB}$ used for out-of-sample extension is defined based on the encoding vectors for the training points. If Y is the encoding matrix for the training points, the $\mathcal{CB} = \{c_q\}_{q=1}^Q$, where $c_q \in \{-1, 1\}^Q$, is defined by the unique rows of Y (i.e. from identical rows of Y one selects one row). The score variables evaluated at the test set $\mathcal{D}^{\text{test}} = \{x_i\}_{i=1}^{n_{\text{test}}}$ become:

$$e_{\text{test}}^{(\ell)} = \Phi_{\text{RFF,test}} w^{(\ell)} + b^{(\ell)} 1_{n_{\text{test}}} \; \ell = 1, \ldots, Q, \tag{20}$$

where $\Phi_{\text{RFF,test}} = [z(x_1), \ldots, z(x_{n_{\text{test}}})]^T \in \mathbb{R}^{n_{\text{test}} \times 2D}$. The decoding scheme consists of comparing the binarized score variables for test data points with the codebook $\mathcal{CB}$ and selecting the nearest codeword in terms of Hamming distance. The procedure for the RFF-MSSKSC approach is summarized in Algorithm 1.

Algorithm 1: RFF-MSSKSC model for large scale data

Input: Training data set $\mathcal{D}$, labels Y, tuning parameters γ_1 and γ_2, kernel parameter (if any), test set $\mathcal{D}^{\text{test}} = \{x_i\}_{i=1}^{n_{\text{test}}}$ and codebook $\mathcal{CB} = \{c_q\}_{q=1}^Q$

Output: Class membership of test data points $\mathcal{D}^{\text{test}}$

1. Obtain $2D$-dimensional random Fourier feature map using (16).
2. Compute $\{w^{(\ell)}\}_{\ell=1}^Q$ and the bias term $\{b^{(\ell)}\}_{\ell=1}^Q$ using (19).
3. Estimate the test data projections $\{e_{\text{test}}^{(\ell)}\}_{\ell=1}^Q$ using (20).
4. Binarize the test projections and form the encoding matrix $[\text{sign}(e_{\text{test}}^{(1)}), \ldots, \text{sign}(e_{\text{test}}^{(Q)})]_{n_{\text{test}} \times Q}$ for the test points (Here $e_{\text{test}}^{(\ell)} = [e_{\text{test},1}^{(\ell)}, \ldots, e_{\text{test},n_{\text{test}}}^{(\ell)}]^T$).
5. $\forall i \, (i = 1, \ldots, n_{\text{test}})$, assign x_i to class q^*, where $q^* = \underset{q}{\text{argmin}} \, d_H(e_{\text{test},i}^{\ell}, c_q)$ and $d_H(\cdot, \cdot)$ is the Hamming distance.

A Matlab demo of the algorithm can be downloaded at: https://sites.google.com/site/smkmhr/code-data

***Remark** 4.1:* It should be noted that, in order to have a fair comparison, in our experiments, the three models FS-MSSKSC, RD-MSSKSC and RFF-MSSKSC use explicit feature maps of the same dimension, i.e. $\bar{n} = m = 2D$ in (11), (13) and (19).

V. Numerical Experiments

In this section experimental results on synthetic and real-life datasets taken from UCI machine learning repository[1] [24] and LIBSVM datasets [2] [25] are given. The experiments are performed on a laptop computer with Intel Core i7 CPU and 16 GB RAM under Matlab 2014a.

[1]Available at: http://archive.ics.uci.edu/ml/datasets.html
[2]Available at: http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/

The performance of the proposed methods depends on the choice of the tuning parameters. In this paper for all the experiments the Gaussian RBF kernel is used. The optimal values of the regularization constants γ_1, γ_2 and the kernel bandwidth parameter σ are obtained by evaluating the performance of the model (classification accuracy) on the validation set. A two step procedure which consists of Coupled Simulated Annealing (CSA) [26] initialized with 5 random sets of parameters for the first step and the simplex method [27] for the second step. CSA is used for determining good initial starting values and then the simplex procedure refines our selection, resulting in more optimal tuning parameters.

The performance of the proposed RFF-MSSKSC algorithm on two-moons and two-spirals datasets with 20000 data points are shown in Figure 1. The training set used for the experiments on these two datasets consists of 100 labeled and 10000 unlabeled data points. The size of the real-life data, on which the experiments were conducted, ranges from medium to large and covering both binary and multi-class classification. The classification of these datasets is performed using different number of training labeled and unlabeled data instances. In our experiments, for all the datasets, 20% of the whole data points (at random) is used as test set, and the training set is constructed from the reaming 80% of the data points. In oder to have a realistic setting, the number of unlabeled training points are considered to be p times more than that of labeled training points, where, in our experimeents, depending on the size of the dataset under study, p ranges from 3 to 5. Descriptions of the used datasets can be found in Table I.

TABLE I
Dataset statistics

Dataset	# of data points	# of attributes	# of classes
Magic	19,020	10	2
Adult	48,842	14	2
Shuttle	57,999	9	2
IJCNN	141,691	22	3
Skin	245,057	3	2
Cod-rna	331,152	8	2
Covertype	581,012	54	3
SUSY	5,000,000	18	2

In both FS-MSSKSC and RD-MSSKSC approaches, the prototype vectors are selected via maximization of the Rényi entropy. As in the semi-supervised setting, one often encounters a small numbers of labeled and a large numbers of unlabeled data points, the total number of prototype vectors consists of prototype vectors selected from labeled and unlabeled data points. In particular, the following experimental protocols for the number prototype vectors (PV) are used:

$$PV_L = \begin{cases} n_L & \text{if } n_L < 200 \\ \lceil q_1 \sqrt{n_L} \rceil & \text{otherwise,} \end{cases} \tag{21}$$

where $q_1 \in \mathbb{Q}^+ \backslash \{0\}$. The number of unlabeled prototype vectors as follows:

$$PV_u = \begin{cases} n_{UL} & \text{if } n_{UL} < 500 \\ q_2 \sqrt{n_{UL}} & \text{otherwise,} \end{cases} \tag{22}$$

where $q_2 \in \mathbb{Q}^+ \backslash \{0\}$. For all the experiments in this paper, q_1 and q_2 are set to one. However one may note that q_1, q_2 and p are the user defined parameters and can be designed in accordance with the available memory of the computer and the size of the dataset under study. The obtained results of the proposed RFF-MSSKSC model together with the FS-MSSKSC and RD-MSSKSC approaches [20] are tabulated in Table II. The results reported in Table II, are obtained by averaging over 10 simulation runs.

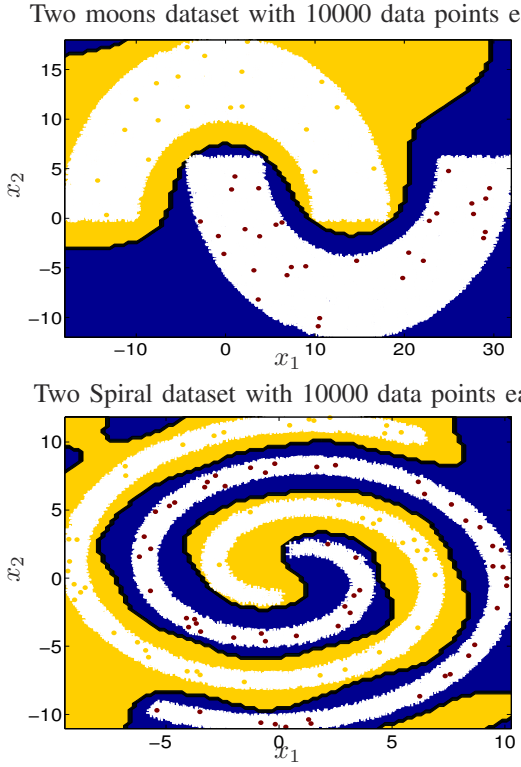

Fig. 1. The performance of the RFF-MSSKSC method on two-moons and two-spirals datasets. In total there are 20000 data points. The dimension of explicit random feature map is set to 300.

Table II shows that for these data one can improve the generalization performance by increasing both labeled and unlabeled data points. In addition, the test accuracy of RFF-MSSKSC and FS-MSSKSC are comparable and also better than that of RD-MSSKSC in most cases. However, thanks to the randomization step involved for constructing the explicit feature map, the proposed RFF-MSSKSC shows significant improvement over the other two approaches in terms of training as well as test computation time without compromising its accuracy on the test set. The fact that for these datasets, the proposed RFF-MSSKSC requires much less training time to produce comparable results, compare to its counterparts, makes it more appealing over the other two approaches for large scale data.

In Fig 2, we examine the performance of the three models (RFF,FS,RD)-MSSKSC on four datasets IJCNN, Cod-rna, Covertype and SUSY with different training set sizes. From Fig 2(a,d,g,j), one cane observe that the test accuracy of the three models improves by increasing the size of the training set (i.e. number of both labeled and unlabeled training data points) for all the datasets. Moreover, the accuracy of RFF-MSSKSC and FS-MSSKSC show a better performance compared to RD-MSSKSC. The required computation time (composed of both training and test stages) of the three models versus the number of training points are shown in Fig 2(b,e,h,k). One can also observe that the RFF-MSSKSC model needs the least amount of training/test time among the other two models. Finally, the test accuracy of the three models versus the required computation time for the above-mentioned four datasets are shown in Fig 2(c,f,i,l), where the proposed RFF-MSSKSC shows a considerably reduced computation times to produce the same or almost comparable level of accuracy with respect to other two approaches. It should also be mentioned that as RD-MSSKSC model does not involve eigen-decomposition step, its training computation times is less than that of FS-MSSKSC.

VI. Conclusions

In this paper, an approach that uses random Fourier features is proposed to make the semi-supervised KSC based algorithm scalable. The proposed model, i.e. RFF-MSSKSC, uses the explicit feature map and solves the semi-supervised optimization problem in the primal. The efficiency and applicability of the proposed method is shown on synthetic and real benchmark datasets. The proposed RFF-MSSKSC model outperforms the Fixed-Size MSSKSC (FS-MSSKSC) and Reduced MSSKSC (RD-MSSKSC) [18] in all cases in terms of training computation times, while the test accuracy of the RFF-MSSKSC is comparable to that of FS-MSSKSC and RD-MSSKSC.

Acknowledgments

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/20072013) / ERC AdG A-DATADRIVE-B (290923). This paper reflects only the authors' views, the Union is not liable for any use that may be made of the contained information; Research Council KUL: GOA/10/09 MaNet, CoE PFV/10/002 (OPTEC), BIL12/11T; PhD/Postdoc grants; Flemish Government: FWO: PhD/Postdoc grants, projects: G.0377.12 (Structured systems), G.088114N (Tensor based data similarity); IWT: PhD/Postdoc grants, projects: SBO POM (100031); iMinds Medical Information Technologies SBO 2014; Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 20122017). Siamak Mehrkanoon is a postdoctoral researcher at KU Leuven, Belgium. Johan Suykens is a full professor at KU Leuven, Belgium.

References

[1] V. Mayer-Schönberger and K. Cukier, *Big data: A revolution that will transform how we live, work, and think*. Houghton Mifflin Harcourt, 2013.

[2] A. J. Smola and B. Schölkopf, "Sparse greedy matrix approximation for machine learning," *17th International Conference on Machine Learning, Stanford, 2000*, pp. 911–918, 2000.

[3] S. Fine and K. Scheinberg, "Efficient SVM training using low-rank kernel representations," *The Journal of Machine Learning Research*, vol. 2, pp. 243–264, 2002.

[4] G. H. Golub and C. F. Van Loan, *Matrix computations*. Johns Hopkins University Press, 2012.

[5] F. R. Bach and M. I. Jordan, "Predictive low-rank decomposition for kernel methods," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 33–40.

[6] S. Kumar, M. Mohri, and A. Talwalkar, "Sampling methods for the Nyström method," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 981–1006, 2012.

[7] K. Zhang, L. Lan, J. T. Kwok, S. Vucetic, and B. Parvin, "Scaling up graph-based semisupervised learning via prototype vector machines," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 3, pp. 444–457, 2015.

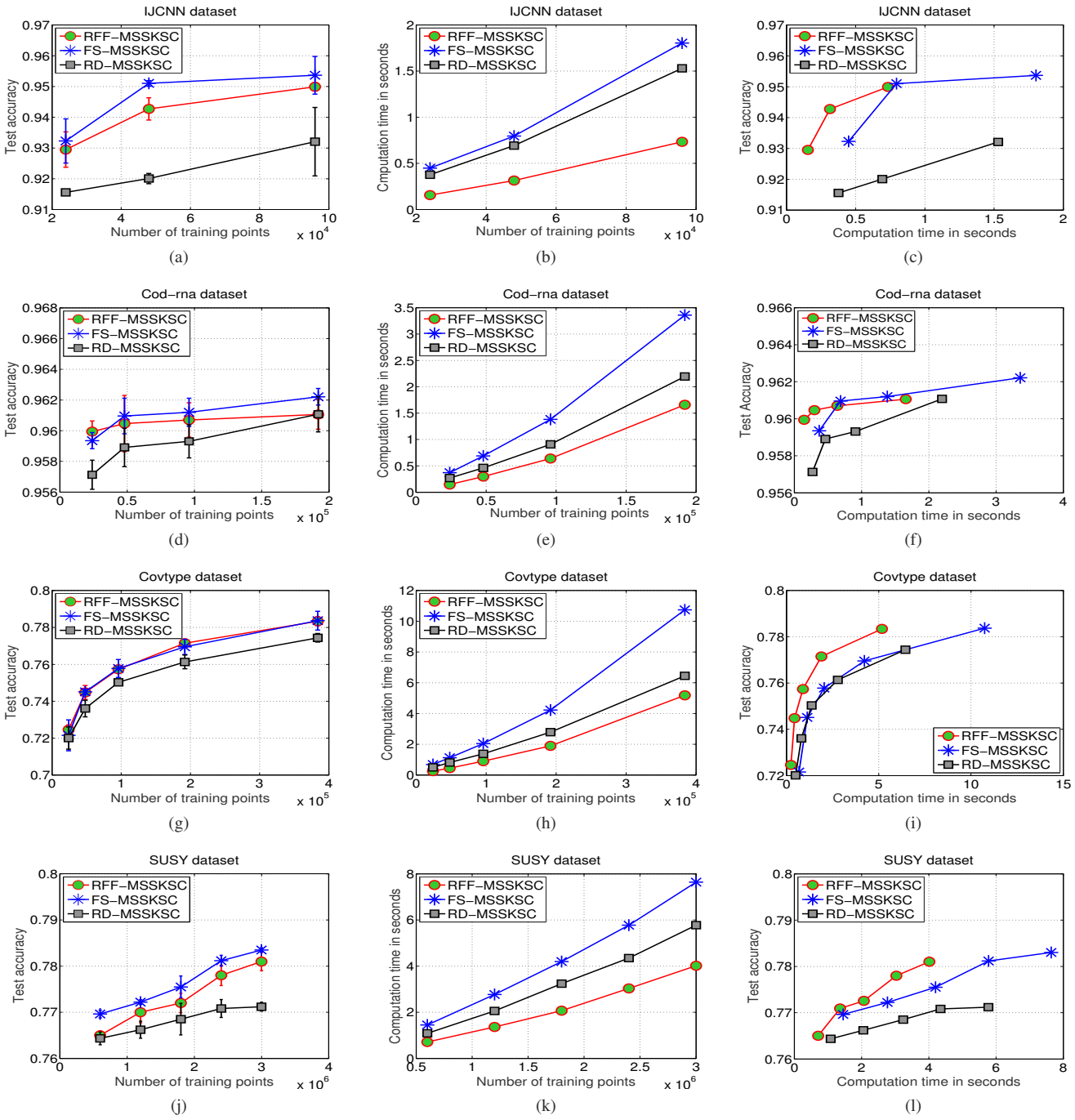

Fig. 2. The performance of the three models (RFF,FS,RD)-MSSKSC on four datasets IJCNN, Cod-rna, Covertype and SUSY. (a,d,g,j) Obtained test accuracy over 10 simulation runs using RF-MSSKSC, FS-MSSKSC and RD-MSSKSC models for the four datasets when different training set sizes are used. (b,e,h,k) Required computation times (composed of training and test stages) versus the number of training data points using RF-MSSKSC, FS-MSSKSC and RD-MSSKSC models for the four datasets. (c,f,i,l) Obtained test accuracy versus elapsed computation times using RF-MSSKSC, FS-MSSKSC and RD-MSSKSC models for the four datasets.

TABLE II
COMPARING THE AVERAGE TEST ACCURACY AND COMPUTATION TIME OF THE PROPOSED RFF-MSSKSC APPROACH WITH THOSE OF FS-MSSKSC AND RD-MSSKSC MODELS [18] ON REAL-LIFE DATASETS OVER 10 SIMULATION RUNS.

					Method			(Training/Test) computation time in seconds		
Dataset	(p, q_1, q_2)	$\mathcal{D}_{\text{tr}}^L$	$\mathcal{D}_{\text{tr}}^U$	$\mathcal{D}_{\text{test}}$	**FS-MSSKSC**	**RD-MSSKSC**	**RFF-MSSKSC**	**FS-MSSKSC**	**RD-MSSKSC**	**RFF-MSSKSC**
Magic	(5,1,1)	1000	5000	3804	0.830	0.816	**0.832**	0.02/0.01	0.02/0.01	**0.009/0.001**
		2000	10000	3804	0.842	0.832	**0.844**	0.03/0.02	0.02/0.01	**0.01/0.01**
Adult	(3,1,1)	4000	12000	9768	**0.845**	**0.845**	0.843	0.05/0.04	0.03/0.04	**0.02/0.02**
		8000	24000	9768	**0.846**	0.845	**0.846**	0.13/0.20	0.10/0.13	**0.08/0.10**
Shuttle	(3,1,1)	4000	12000	11599	0.993	0.988	**0.994**	0.05/0.08	0.04/0.07	**0.03/0.04**
		8000	24000	11599	**0.995**	**0.995**	**0.995**	0.12/0.04	0.11/0.03	**0.08/0.02**
IJCNN	(5,1,1)	4000	20000	28338	**0.935**	0.915	0.929	0.16/0.30	0.13/0.25	**0.06/0.21**
		16000	80000	28338	**0.955**	0.938	0.950	1.31/0.60	1.09/0.52	**0.62/0.33**
Skin	(5,1,1)	8000	40000	49011	**0.997**	**0.997**	**0.997**	0.18/0.29	0.16/0.25	**0.14/0.10**
		32000	160000	49011	**0.998**	**0.998**	**0.998**	1.50/0.76	1.17/0.53	**0.85/0.40**
Cod-rna	(5,1,1)	8000	40000	66230	0.959	0.958	**0.960**	0.27/0.40	0.17/0.36	**0.15/0.14**
		32000	160000	66230	**0.962**	0.961	0.961	2.26/1.21	1.43/0.99	**1.32/0.33**
Covertype	(5,1,1)	8000	40000	116202	0.732	0.731	**0.742**	0.25/0.85	0.17/0.67	**0.15/0.39**
		64000	320000	116202	**0.781**	0.772	**0.781**	7.76/3.09	4.70/2.76	**4.60/0.94**
SUSY	(2,0.05,0.05)	500000	1000000	1000000	**0.771**	0.762	0.769	2.05/1.61	1.41/1.26	**1.24/0.71**
		1000000	2000000	1000000	**0.783**	0.771	0.781	5.87/2.61	4.12/1.76	**3.72/0.78**

[8] C. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Proceedings of the 14th Annual Conference on Neural Information Processing Systems*, no. EPFL-CONF-161322, 2001, pp. 682–688.

[9] X. Zhu, "Semi-supervised learning literature survey," *Computer Science, University of Wisconsin-Madison*, 2006.

[10] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *The Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.

[11] S. Mehrkanoon, C. Alzate, R. Mall, R. Langone, and J. A. K. Suykens, "Multi-class semi-supervised learning based upon kernel spectral clustering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 4, pp. 720–733, 2015.

[12] S. Mehrkanoon, O. M. Agudelo, and J. A. K. Suykens, "Incremental multi-class semi-supervised clustering regularized by Kalman filtering," *Neural Networks*, vol. 71, pp. 88–104, 2015.

[13] S. Mehrkanoon and J. A. K. Suykens, "Multi-label semi-supervised learning using regularized kernel spectral clustering," in *Proc. of IEEE International Joint Conference on Neural Networks (WCCI-IJCNN), Vancouver, Canada*, 2016, pp. 4009–4016.

[14] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-supervised learning*. MIT press Cambridge, 2006, vol. 2.

[15] V. Sindhwani and S. S. Keerthi, "Large scale semi-supervised linear SVMs," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2006, pp. 477–484.

[16] K. Zhang, J. T. Kwok, and B. Parvin, "Prototype vector machine for large scale semi-supervised learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 1233–1240.

[17] W. Liu, J. He, and S.-F. Chang, "Large graph construction for scalable semi-supervised learning," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 679–686.

[18] S. Mehrkanoon and J. A. K. Suykens, "Large scale semi-supervised learning using KSC based model," in *Proc. of IEEE International Joint Conference on Neural Networks (IJCNN), Beijing, China*, 2014, pp. 4152–4159.

[19] C. Alzate and J. A. K. Suykens, "Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 2, pp. 335–347, 2010.

[20] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, *Least squares support vector machines*. Singapore: World Scientific Pub. Co., 2002.

[21] K. De Brabanter, J. De Brabanter, J. A. K. Suykens, and B. De Moor, "Optimized fixed-size kernel models for large data sets," *Computational Statistics & Data Analysis*, vol. 54, no. 6, pp. 1484–1504, 2010.

[22] M. Girolami, "Orthogonal series density estimation and the kernel eigenvalue problem," *Neural Computation*, vol. 14, no. 3, pp. 669–688, 2002.

[23] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in neural information processing systems*, 2007, pp. 1177–1184.

[24] A. Asuncion and D. J. Newman, "UCI machine learning repository," 2007.

[25] C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, pp. 27:1–27:27, 2011.

[26] S. Xavier-De-Souza, J. A. K. Suykens, J. Vandewalle, and D. Bollé, "Coupled simulated annealing," *IEEE Trans. Sys. Man Cyber. Part B*, vol. 40, no. 2, pp. 320–335, Apr. 2010.

[27] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The computer journal*, vol. 7, no. 4, pp. 308–313, 1965.