2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Large scale semi-supervised learning using KSC based model

Siamak Mehrkanoon and Johan A. K. Suykens

***Abstract*— Often in practice one deals with a large amount of unlabeled data, while the fraction of labeled data points will typically be small. Therefore one prefers to apply a semi-supervised algorithm, which uses both labeled and unlabeled data points in the learning process, to have a better performance. Considering the large amount of unlabeled data, making a semi-supervised algorithm scalable is an important task. In this paper we adopt a recently proposed multi-class semi-supervised KSC based algorithm (MSS-KSC) and make it scalable by means of two different approaches. The first one is based on the Nyström approximation method which provides a finite dimensional feature map that can then be used to solve the optimization problem in the primal. The second approach is based on the reduced kernel technique that solves the problem in the dual by reducing the dimensionality of the kernel matrix to a rectangular kernel. Experimental results demonstrate the scalability and efficiency of the proposed approaches on real datasets.**

I. Introduction

In practice one needs to address the issue of scalability to deal with vast amounts of data. In many applications, ranging from data mining to machine perception, obtaining the labels of input data is often difficult and expensive. Therefore in many cases one encounters a large amount of unlabeled data while the labeled data are rare. Semi-supervised learning is a framework in machine learning that aims at learning from both labeled and unlabeled data points [1]. Using unlabeled data together with labeled data often gives better results than using the labeled data alone. Many semi-supervised algorithms perform well on relatively small problems, (see [2] and references therein), but they do not scale well when deal with large scale data. Therefore turning semi-supervised learning algorithms into practice is important. For instance a family of semi-supervised linear support vector classifiers for large data sets is introduced in [3].

Most of the developed semi-supervised approaches attempt to improve the performance by incorporating the information from either the unlabeled or labeled part. Among them are 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) [4], a state of art method in semi-supervised classification, is one of the graph based methods which provide a natural out-of-sample extension.

Kernel spectral clustering (KSC) is an unsupervised algorithm introduced in [5]. The primal problem of the kernel spectral clustering is formulated as a weighted kernel PCA.

The authors are with the Department of Electrical Engineering ESAT-STADIUS, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium (email: {siamak.mehrkanoon, johan.suykens}@esat.kuleuven.be).

In [6] the out-of-sample extension property of KSC is used to introduce sparsity to the KSC model for large-scale data sets. The authors in [7] have extended the kernel spectral clustering to semi-supervised learning by incorporating the information of labeled data points in the learning process. Recently Mehrkanoon *et al.* [8] proposed a multi-class semi-supervised algorithm (MSS-KSC) where KSC is used as a core model. The available side-information (labels) is incorporated to the core model through a regularization term.

In the MSS-KSC approach, one needs to solve a linear system of equations to obtain the model parameters. Therefore with n number of training points, the algorithm has $\mathcal{O}(n^3)$ training complexity with naive implementations.

It is the purpose of this paper to make the recently proposed MSS-KSC algorithm of [8] scalable. To this end, we propose two possible schemes:

- The first approach, which will be referred to as Fixed-Size MSS-KSC (FS-MSS-KSC), is based on the Nyström approximation and the primal-dual formulation of the MSS-KSC. 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 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. The second approach will be referred to as Reduced MSS-KSC (RD-MSS-KSC) approach.

This paper is organized as follows. In Section II, a brief review of binary kernel spectral clustering is given. Section III briefly reviews the Nyström method for approximating the finite dimensional feature map. In Section IV the Fixed-size MSS-KSC approach for large scale problem is formulated. Section V, introduces the Reduced MSS-KSC approach for large scale problems. In section VI model selection aspects are discussed. Simulation results are presented in Section VII to show the performance of the proposed algorithms.

II. Brief overview of KSC

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 [5]:

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

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

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 $\varphi(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^h$ 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:

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

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 $VP_v\Omega$ 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. Approximation to the feature map

In order to handle large data sets the so called fixed-size approach, where the feature map is approximated by the Nyström method [9], [10], is introduced in [11] and has been applied in [12], [13]. In what follows, we briefly summarize the fixed-size approach.

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. Consider the Fredholm integral equation of the first kind:

$$\int_C K(x, x_j)\phi_i(x)p(x)dx = \lambda_i\phi_i(x_j) \tag{3}$$

where C is a compact subset of $\mathbb{R}^d$. The approximation of the eigenfunction $\phi_i(x)$ in (3) can be obtained by the Nyström method which applies a quadrature rule for discretizing the left-hand side of (3). This will lead to the eigenvalue problem [9]:

$$\frac{1}{n}\sum_{k=1}^n K(x_k, x_j)u_{ik} = \lambda_i^{(s)} u_{ij} \tag{4}$$

where the eigenvalues λ_i and eigenfunctions ϕ_i from the continuous problem (3) can be approximated by the sample eigenvalues $\lambda_i^{(s)}$ and eigenvectors u_i. Therefore, the i-th component of the n-dimensional feature map $\hat{\varphi} : \mathbb{R}^d \rightarrow \mathbb{R}^n$, for any point $x \in \mathbb{R}^d$, can be obtained as follows:

$$\hat{\varphi}_i(x) = \frac{1}{\lambda_i^{(s)}}\sum_{k=1}^n u_{ki}\, K(x_k, x) \tag{5}$$

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 $m \ll n$ whose elements are selected using an entropy based criterion. In this case, the m-dimensional feature map $\hat{\varphi} : \mathbb{R}^d \rightarrow \mathbb{R}^m$ can be approximated as follows:

$$\hat{\varphi}(x) = [\hat{\varphi}_1(x), \ldots, \hat{\varphi}_m(x)]^T \tag{6}$$

where

$$\hat{\varphi}_i(x) = \frac{1}{\lambda_i^{(s)}}\sum_{k=1}^m u_{ki}\, K(x_k, x), i = 1, \ldots, m \tag{7}$$

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.

IV. Fixed-Size MSS-KSC for large scale datasets

In this section, first the Fixed-Size MSS-KSC approach is formulated in the primal and then in subsection IV.B derivation of the finite dimensional feature map used in the proposed FS-MSS-KSC is explained.

A. Formulation of the method

Consider training data points

$$\mathcal{D} = \{\underbrace{x_1, ..., x_{n_u}}_{\substack{Unlabeled \\ (\mathcal{D}_U)}}, \underbrace{x_{n_u+1}, .., x_n}_{\substack{Labeled \\ (\mathcal{D}_L)}}\},$$

where $\{x_i\}_{i=1}^n \in \mathbb{R}^d$. The first n_u data points do not have labels whereas the last $n_L = n - n_u$ 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:

$$Y_{ij} = \begin{cases} +1 & \text{if the } i\text{th point belongs to the } j\text{th class} \\ -1 & \text{otherwise.} \end{cases} \tag{8}$$

The information of the labeled data is incorporated to the kernel spectral clustering (1) 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. The formulation of Multi-class semi-supervised KSC (MSS-KSC) described in [8] in primal is given as follows:

$$\begin{aligned} \min_{w^{(\ell)}, b^{(\ell)}, e^{(\ell)}} \quad & \frac{1}{2}\sum_{\ell=1}^{Q} {w^{(\ell)}}^T w^{(\ell)} - \frac{\gamma_1}{2}\sum_{\ell=1}^{Q} {e^{(\ell)}}^T V e^{(\ell)} + \\ & \frac{\gamma_2}{2}\sum_{\ell=1}^{Q}(e^{(\ell)} - c^{(\ell)})^T A (e^{(\ell)} - c^{(\ell)}) \\ \text{subject to} \quad & e^{(\ell)} = \Phi w^{(\ell)} + b^{(\ell)} 1_n, \ \ell = 1, \ldots, Q, \end{aligned} \tag{9}$$

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

$$C = [c^{(1)}, \ldots, c^{(Q)}]_{n \times Q} = \left[\frac{0_{n_u \times Q}}{Y}\right]_{n \times Q}, \tag{10}$$

where $0_{n_u \times Q}$ is a zero matrix of size $n_u \times Q$ and Y is defined as previously. The matrix A is defined as follows:

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

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.

Since in Equation (9) the feature map φ is not explicitly known, one uses the kernel trick and solves the problem in the dual. But as it has been shown in [8] in the dual one has to solve a linear system of size n (number of data points). Therefore for large scale data, it is not appropriate to solve the problem in the dual. In what follows we show how one can use the approximation of the feature map (explained in section III) to solve the problem in primal. Given the finite dimensional (m-dimensional) approximation to the feature map, i.e.

$$\hat{\Phi} = [\hat{\varphi}(x_1), \ldots, \hat{\varphi}(x_n)]^T \in \mathbb{R}^{n \times m} \tag{11}$$

one can rewrite the above optimization problem as an unconstrained optimization problem and solve it in primal:

$$\begin{aligned} \min_{w^{(\ell)}, b^{(\ell)}} J(w^{(\ell)}, b^{(\ell)}) = & \frac{1}{2}\sum_{\ell=1}^{Q} {w^{(\ell)}}^T w^{(\ell)} - \\ & \frac{\gamma_1}{2}\sum_{\ell=1}^{Q}(\hat{\Phi} w^{(\ell)} + b^{(\ell)} {1_N}^T)^T V (\hat{\Phi} w^{(\ell)} + b^{(\ell)} 1_N) + \\ & \frac{\gamma_2}{2}\sum_{\ell=1}^{Q}(c^{(\ell)} - \hat{\Phi} w^{(\ell)} + b^{(\ell)} 1_n)^T A (c^{(\ell)} - \hat{\Phi} w^{(\ell)} + b^{(\ell)} 1_n) \end{aligned} \tag{12}$$

where the matrix C is defined as previously.

***Lemma** 4.1:* Given a finite dimensional (m-dimensional) approximation to the feature map $\hat{\Phi}$ and regularization constants $\gamma_1, \gamma_2 \in \mathbb{R}^+$, the solution to (12) is obtained by solving the following linear system:

$$\left[\begin{array}{c} w^{(\ell)} \\ b^{(\ell)} \end{array}\right] = \left(\Phi_e^T R \Phi_e + I_{(m+1)}\right)^{-1} \gamma_2 \Phi_e^T c^{(\ell)}, \ell = 1, \ldots, Q, \tag{13}$$

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 $(m+1) \times (m+1)$.

Proof: Taking the derivative of the cost function J with respect to $w^{(\ell)}$ and $b^{(\ell)}$ yields:

$$\begin{cases} \frac{\partial \mathcal{J}}{\partial w^{(\ell)}} = 0 \rightarrow \\ (I + \hat{\Phi}^T R \hat{\Phi}) w^{(\ell)} + \hat{\Phi}^T R 1_n b^{(\ell)} = \gamma_2 \hat{\Phi}^T c^{(\ell)}, \ \ell = 1, \ldots, Q, \\ \frac{\partial \mathcal{L}}{\partial b^{(\ell)}} = 0 \rightarrow \\ 1_n^T R \hat{\Phi} w^{(\ell)} + (1_n^T R 1_n) b^{(\ell)} = \gamma_2 1_n^T c^{(\ell)}, \ \ell = 1, \ldots, Q, \end{cases} \tag{14}$$

which then by using some algebraic manipulation can be rewritten as in (13). ■

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:

$$e_{\text{test}}^{(\ell)} = \hat{\Phi}_{\text{test}} w^{(\ell)} + b^{(\ell)} 1_{n_{\text{test}}} \ \ell = 1, \ldots, Q, \tag{15}$$

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 the codebook $\mathcal{CB}$ and selecting the nearest codeword in terms of Hamming distance. The procedure for the Fixed-Size MSS-KSC approach is summarized in Algorithm 1.

B. Subsample selection for Nyström approximation

We aim at using an m-dimensional approximation to the feature map φ. Therefore as it is explained in section III, one needs to select a subset of fixed size m from a

Algorithm 1: Fixed-size MSS-KSC approach 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 Select m prototype vectors (small working set) using quadratic Rényi entropy criterion [14]. (see section IV. B)
2 Obtain the m-dimensional approximation of the feature map (11) by means of Nyström approximation (7).
3 Compute $\{w^{(\ell)}\}_{\ell=1}^{Q}$ and the bias term $\{b^{(\ell)}\}_{\ell=1}^{Q}$ using (13).
4 Estimate the test data projections $\{e_{\text{test}}^{(\ell)}\}_{\ell=1}^{Q}$ using (15).
5 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 $e_{\text{test}}^{(\ell)} = [e_{\text{test},1}^{(\ell)}, \ldots, e_{\text{test},n_{\text{test}}}^{(\ell)}]^T$).
6 $\forall i\,(i = 1, \ldots, n_{\text{test}})$, 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.

pool of training points of size n. Since the training set is composed of labeled and unlabeled data points, we select a subset (of size m) such that it consists of m_1 and m_2 data points from labeled and unlabeled training data points. ($m = m_1 + m_2$). As it has been motivated in [11], the Rényi entropy criterion [14] is used, twice only, to select m_1 points from the labeled and m_2 points from the unlabeled training data. Once the subset is available, the m-dimensional feature map is obtained using equation (7).

V. Reduced MSS-KSC for large scale datasets

For large-scale problems, the difficulty of solving the MSS-KSC formulation (9) in the dual results from the huge kernel matrix which cannot be stored into memory. The authors in [15] proposed to restrict the number of support vectors by solving the reduced support vector machines (RSVM) for classification problem. The reduced kernel technique is utilized to reduce the $n \times n$ dimensionality of the kernel Ω to a much smaller $n \times \bar{n}$ dimensionality. Here $\bar{n}$ is the size of a randomly selected subset of training data considered as candidates of support vectors. A smaller matrix then can be stored into memory.

In what follows, we apply the reduced kernel technique described in [15] to the MSS-KSC formulation (9) in order to make it scalable. Suppose the matrix of training data points which includes both labeled and unlabeled samples is denoted by:

$$X = [x_1, \ldots, x_n]^T \in \mathbb{R}^{n \times d}.$$

Let us start with a linear kernel and reformulate (9) as follows:

$$\begin{aligned} \min_{w^{(\ell)}, b^{(\ell)}, e^{(\ell)}} \quad & \frac{1}{2}\sum_{\ell=1}^{Q}\left({w^{(\ell)}}^T w^{(\ell)} + (b^{(\ell)})^2\right) - \frac{\gamma_1}{2}\sum_{\ell=1}^{Q} {e^{(\ell)}}^T V e^{(\ell)} \\ & + \frac{\gamma_2}{2}\sum_{\ell=1}^{Q}(e^{(\ell)} - c^{(\ell)})^T A (e^{(\ell)} - c^{(\ell)}) \\ \text{subject to} \quad & e^{(\ell)} = X w^{(\ell)} + b^{(\ell)} 1_n,\ \ell = 1, \ldots, Q, \end{aligned} \tag{16}$$

where here the bias term is also penalized just to make the subsequent derivations simpler. Setting the gradient of the associated Lagrangian of (16) with respect to $w^{(\ell)}$ to zero gives the following KKT condition:

$$w^{(\ell)} = X^T \alpha^{(\ell)}, \tag{17}$$

where $\alpha^{(\ell)}$ are the Lagrange multipliers associated with the equality constraint of (16). By replacing the primal variables $w^{(\ell)}$ from (17) one obtains:

$$\begin{aligned} \min_{\alpha^{(\ell)}, b^{(\ell)}, e^{(\ell)}} \quad & \frac{1}{2}\sum_{\ell=1}^{Q}\left({\alpha^{(\ell)}}^T \alpha^{(\ell)} + (b^{(\ell)})^2\right) - \frac{\gamma_1}{2}\sum_{\ell=1}^{Q} {e^{(\ell)}}^T V e^{(\ell)} \\ & + \frac{\gamma_2}{2}\sum_{\ell=1}^{Q}(e^{(\ell)} - c^{(\ell)})^T A (e^{(\ell)} - c^{(\ell)}) \\ \text{subject to} \quad & e^{(\ell)} = X X^T \alpha^{(\ell)} + b^{(\ell)} 1_n,\ \ell = 1, \ldots, Q, \end{aligned} \tag{18}$$

where the objective function is modified to have the L2 norm regularization of the problem variables $\alpha^{(\ell)}, b^{(\ell)}, e^{(\ell)}$. Following the lines of [15] one can now replace the linear kernel matrix XX^T by a nonlinear kernel matrix with elements $\Omega_{ij} = K(x_i, x_j)$ to obtain the following optimization problem:

$$\begin{aligned} \min_{\alpha^{(\ell)}, b^{(\ell)}, e^{(\ell)}} \quad & \frac{1}{2}\sum_{\ell=1}^{Q}\left({\alpha^{(\ell)}}^T \alpha^{(\ell)} + (b^{(\ell)})^2\right) - \frac{\gamma_1}{2}\sum_{\ell=1}^{Q} {e^{(\ell)}}^T V e^{(\ell)} \\ & + \frac{\gamma_2}{2}\sum_{\ell=1}^{Q}(e^{(\ell)} - c^{(\ell)})^T A (e^{(\ell)} - c^{(\ell)}) \\ \text{subject to} \quad & e^{(\ell)} = \Omega \alpha^{(\ell)} + b^{(\ell)} 1_n,\ \ell = 1, \ldots, Q. \end{aligned} \tag{19}$$

***Lemma** 5.1:* Given regularization constants $\gamma_1, \gamma_2 \in \mathbb{R}^+$, the solution to (19) is obtained as follows:

$$\left(R^{-1} + GG^T\right)\beta^{(\ell)} = R\gamma_2 c^{(\ell)}, \ell = 1, \ldots, Q, \tag{20}$$

where $R = \gamma_2 A - \gamma_1 V$ is a diagonal matrix and $G = [\Omega, 1_n]$. $\beta^{(\ell)} = [\beta_1^{(\ell)}, \ldots, \beta_n^{(\ell)}]^T$ are the Lagrange multipliers.

Proof: The Lagrangian of the constrained optimization problem (19) becomes:

$$\begin{aligned} &\mathcal{L}(\alpha^{(\ell)}, b^{(\ell)}, e^{(\ell)}, \beta^{(\ell)}) = \frac{1}{2}\sum_{\ell=1}^{Q}\left({\alpha^{(\ell)}}^T \alpha^{(\ell)} + (b^{(\ell)})^2\right) - \\ &\frac{\gamma_1}{2}\sum_{\ell=1}^{Q} {e^{(\ell)}}^T V e^{(\ell)} + \frac{\gamma_2}{2}\sum_{\ell=1}^{Q}(e^{(\ell)} - c^{(\ell)})^T A (e^{(\ell)} - c^{(\ell)}) + \\ &\sum_{\ell=1}^{Q} {\beta^{(\ell)}}^T \left(e^{(\ell)} - \Omega\alpha^{(\ell)} - b^{(\ell)} 1_n\right), \end{aligned}$$

where $\beta^{(\ell)}$ is the vector of Lagrange multipliers. Then the Karush-Kuhn-Tucker (KKT) optimality conditions are as follows,

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial e^{(\ell)}} = 0 \rightarrow e^{(\ell)} = R^{-1}\Big(\gamma_2 A\, c^{(\ell)} - \beta_\ell\Big),\ \ell = 1,\ldots,Q, \\ \frac{\partial \mathcal{L}}{\partial b^{(\ell)}} = 0 \rightarrow b^{(\ell)} = 1_n^T \beta^{(\ell)},\ \ell = 1,\ldots,Q, \\ \frac{\partial \mathcal{L}}{\partial \alpha^{(\ell)}} = 0 \rightarrow \alpha^{(\ell)} = \Omega^T \beta^{(\ell)}, \ell = 1,\ldots,Q, \\ \frac{\partial \mathcal{L}}{\partial \beta^{(\ell)}} = 0 \rightarrow \Omega\alpha^{(\ell)} + b^{(\ell)} 1_n = e^{(\ell)},\ \ell = 1,\ldots,Q, \end{cases} \tag{21}$$

where R is defined as previously. Elimination of the primal variables $\alpha^{(\ell)}, e^{(\ell)}$, results in the following equation

$$\Big(R^{-1} + GG^T\Big)\beta^{(\ell)} = R\gamma_2 c^{(\ell)}, \ell = 1,\ldots,Q, \tag{22}$$

with G defined as previously. ■

Obviously for large scale data, still matrix G is of size $n \times n$ which is problematic. Therefore here the reduced kernel technique can be used to overcome this issue by reducing the $n \times n$ dimensionality of kernel 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 X. In this paper the subset is selected using a Rényi entropy based criterion [14]). If one works with the reduced kernel $\bar{\Omega}$ in the primal optimization problem (19), then by using the Sherman-Morrison-Woodbury formula [16], the solution in the dual can be obtained as follows:

$$\beta^{(\ell)} = \left[I_n - R\bar{G}\Big(I_{\bar{n}+1} + \bar{G}^T R \bar{G}\Big)^{-1} \bar{G}^T\right]\gamma_2 c^{(\ell)}, \ell = 1,\ldots,Q, \tag{23}$$

where $\bar{G} = [\bar{\Omega}, 1_n] \in \mathbb{R}^{n \times (\bar{n}+1)}$ and I_n is the identity matrix. The expression (23) involves the inversion of a small matrix of order $(\bar{n}+1) \times (\bar{n}+1)$. After obtaining the $\beta^{(\ell)}$, the score variables evaluated at the test set $X^{\text{test}} = \{x_i\}_{i=1}^{n_{\text{test}}}$ become:

$$\begin{aligned} e_{\text{test}}^{(\ell)} &= \bar{\Omega}^{\text{test}}\, \alpha^{(\ell)} + b^{(\ell)} 1_{n_{\text{test}}} \\ &= \left[\bar{\Omega}^{\text{test}}\, \bar{\Omega}^T\right]\beta^{(\ell)} + b^{(\ell)} 1_{n_{\text{test}}},\ \ell = 1,\ldots,Q, \end{aligned} \tag{24}$$

where $\bar{\Omega}^{\text{test}}_{ij} = K(x_i, x_j)$ 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 the codebook $\mathcal{CB}$ and selecting the nearest codeword in terms of Hamming distance. The procedure for Reduced MSS-KSC is summarized in Algorithm 2.

***Remark** 5.1:* Without loss of generality, in our experiments we set $\bar{n}$ (in Algorithm 2) equal to the number of prototype vectors, i.e. m, used in Algorithm 1.

***Remark** 5.2:* Based on the given formulations in section IV and V, the following differences between the Reduced and Fixed-size MSS-KSC can be observed:

In the Fixed-Size MSS-KSC approach:

- One relies on the eigen-decomposition of the kernel matrix (associated with the prototype vectors) to approximate the feature map.
- The solution vector $w^{(\ell)}$ obtained by Fixed-size MSS-KSC has the same dimension as the number of prototype vectors.
- One solves the problem in the primal.
- Computational complexity, neglecting lower order terms, for solving linear system (13) is $\mathcal{O}(2nm^2 + 2mn + 2m^3 + m^2)$ with $m \ll n$. (The complexity of calculating the Nyström approximation $\mathcal{O}(m^3 + m^2 n)$ is also included).

In the Reduced MSS-KSC approach:

- 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 solution vector $\beta^{(\ell)}$ obtained by Reduced MSS-KSC has the same dimension as the number of training points.
- One solves the problem in dual.
- Computational complexity, neglecting lower order terms, for solving linear system (23) is $\mathcal{O}(nm^2 + 3mn + m^3 + m^2)$ with $m \ll n$.

Algorithm 2: Reduced MSS-KSC approach for large scale data

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

Output: Class membership of test data points X^{test}

1. Select a subset matrix $\bar{X} \in \mathbb{R}^{\bar{n} \times d}$ from the original training data matrix $X \in \mathbb{R}^{n \times d}$ using Rényi entropy based criterion [14]).
2. Solve the linear system (23) to obtain $\{\beta^{(\ell)}\}_{\ell=1}^{Q}$ and compute the bias term $\{b^{(\ell)}\}_{\ell=1}^{Q}$ using the second equation of the KKT condition (21).
3. Estimate the test data projections $\{e_{\text{test}}^{(\ell)}\}_{\ell=1}^{Q}$ using (24).
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.

VI. Model selection

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) [17] initialized with 5 random sets of parameters for the first step and the simplex method [18] 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.

Noting that both labeled and unlabeled data points are involved in the learning process, it is natural to have a model selection criterion that makes use of both labeled and unlabeled data points. Therefore as in [7], [19] we use a criterion which is an affine combination of classification accuracy and the clustering performance of the underlying model. The model selection criterion can be expressed as follows:

$$\underset{\gamma_1,\gamma_2,\sigma}{\operatorname{argmax}} \kappa\, CLP(\gamma_1,\gamma_2,\gamma_3,\sigma) + (1-\kappa) Acc(\gamma_1,\gamma_2,\gamma_3,\sigma)$$

where *CLP* and *Acc* stand for clustering performance and classification accuracy respectively. $\kappa \in [0,1]$ is a user-defined parameter that controls the trade-off between the importance given to unlabeled and labeled samples. A common approach for evaluation of clustering results is to use cluster validity indices [20], [21], [22]. Any internal clustering validity approach such as Silhouette index [23], Davies-Bouldin index (DB) or BLF [5] can be utilized. In this paper we explored the BLF and Silhouette indices and the result of the one with highest accuracy (on the validation set) is reported.

VII. 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 4 GB RAM under Matlab 2012a.

The performance of the proposed FS-MSS-KSC algorithm on two moons dataset with 4000 data points is shown in Figure 1. The selected prototype vectors are depicted by circles.

For the real datasets the size of the data on which the experiments were conducted ranges from small to large and covering both binary and multi-class classification. The amount of labeled data points used in the learning process, depending on the size of the dataset, ranges from 1% to 40% of the remaining data points (i.e. test set is not included).

Descriptions of the used datasets from [24] and [25] can be found in Table I. For Ecoli and Covertype datasets we merge some of the classes in order to avoid unbalanced classes. In both Fixed-Size MSS-KSC and Reduced MSS-KSC approaches the prototype vectors (small working set) were selected via maximization of the Rényi entropy. The total amount of prototype vectors consists of prototype vectors selected from labeled and unlabeled data points. Noting that in the semi-supervised setting one usually encounters a small amount of labeled and a large amount of unlabeled data points, in our experiments, for the labeled data points the number of the prototype vectors is set as follows:

$$PV_L = \begin{cases} n_L & \text{if } n_L < 200 \\ \lceil q_1 \sqrt{n_L} \rceil & \text{otherwise,} \end{cases} \tag{25}$$

where $q_1 \in \mathbb{Q}^+ \backslash \{0\}$. For all the experiments q_1 is set to one. For the unlabeled data points if its number is small (less than 1000) then the number of the prototype vectors is set as follows:

$$PV_u = \begin{cases} n_u & \text{if } n_u < 500 \\ \lceil \sqrt{n_u} \rceil & \text{otherwise.} \end{cases} \tag{26}$$

In case the amount of unlabeled data points is huge, first we randomly select a fraction of them of size $n_u^{\text{new}} = \lceil p\, n_L \rceil$, where $p \in \mathbb{N}$, for training set and then choose the number of prototype vectors from the new set of unlabeled data points as follows:

$$PV_u = \begin{cases} \lceil n_u^{\text{new}} \rceil & \text{if } \lceil n_u^{\text{new}} \rceil < 500 \\ \lceil q_2 \sqrt{n_u^{\text{new}}} \rceil & \text{otherwise,} \end{cases} \tag{27}$$

where $q_2 \in \mathbb{Q}^+ \backslash \{0\}$. It should be noted that q_1, q_2 and p are the user defined parameters that can be designed in accordance with the available memory of the computer that is being used to conduct the experiments. The obtained results of the proposed (Fixed-Size and Reduced) MSS-KSC approaches together with the Fixed-Size implementation of the LSSVM approach [11] are tabulated in Table II. The results reported in Table II, are obtained by averaging over 10 simulation runs with $\kappa = 0.25$ used in the model selection criterion. For the LapSVMp approach, we tuned the kernel parameter and γ_A with respect to the accuracy on the validation set. The remaining parameters, i.e. γ_I and NN (the number of neighbors), are set to their default values ($\gamma_I = 1$ and $NN = 6$).

TABLE I
Dataset statistics

Dataset	# of data points	# of attributes	# of classes
Iris	154	4	3
Spect	267	21	2
Heart	270	13	2
Ecoli	336	7	5
Pima-Indian	768	8	2
Spambase	4597	57	2
Satimage	6435	36	6
Ring	7400	20	2
Magic	19020	10	2
Cod-rna	331152	8	2
Covertype	581012	54	3

Table II shows that for these data one can improve the generalization performance by incorporating unlabeled data points into the learning process. It should be noted that the FS-LSSVM is a supervised algorithm that uses only the labeled training points. The training computation times for the algorithms used to obtain the results of Table II are then reported in Table III. These results are expected since the FS-LSSVM does not use unlabeled data in the training process therefore it is the fastest one. The FS-MSS-KSC requires to

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

TABLE II

THE AVERAGE TEST ACCURACY AND THE STANDARD DEVIATION OF THE PROPOSED FIXED-SIZE, REDUCED MSS-KSC APPROACHES AND FIXED-SIZE LSSVM [11] METHOD ON REAL DATASETS OVER 10 SIMULATION RUNS.

Dataset	q_2/p	n_L/n_u (% of Labeled data)	$n_L^{\text{validation}}/n_u^{\text{validation}}$	$\mathcal{D}^{\text{test}}$(%)	PV_L/PV_u	Method			
						FS-MSS-KSC	**RD-MSS-KSC**	**LapSVMp**	**FS-LSSVM**
Heart	1/1	19/76 (20%)	19/75	81 (30%)	19/76	$\mathbf{0.803 \pm 0.05}$	0.795 ± 0.05	0.761 ± 0.001	0.759 ± 0.05
Pima-Indian	1/1	54/215 (20%)	54/215	230 (30%)	54/215	0.740 ± 0.02	0.746 ± 0.02	$\mathbf{0.748 \pm 0.001}$	0.729 ± 0.03
Spect	1/1	19/75 (20%)	19/74	80 (30%)	19/75	0.832 ± 0.07	$\mathbf{0.838 \pm 0.02}$	0.821 ± 0.01	0.825 ± 0.03
Iris	1/1	24/36 (40%)	24/36	30 (20%)	24/36	0.946 ± 0.05	$\mathbf{0.960 \pm 0.02}$	0.938 ± 0.13	0.601 ± 0.05
Ecoli	1/1	54/81 (40%)	54/80	67 (20%)	54/81	0.746 ± 0.03	0.740 ± 0.04	$\mathbf{0.748 \pm 0.06}$	0.468 ± 0.03
Satimage	1/1	1030/1030 (40%)	1030/1030	1287 (20%)	33/33	$\mathbf{0.864 \pm 0.006}$	0.831 ± 0.009	0.834 ± 0.007	0.325 ± 0.08
Ring	1/1	592/592 (20%)	592/592	1480 (20%)	25/25	$\mathbf{0.975 \pm 0.005}$	0.974 ± 0.005	0.972 ± 0.006	0.968 ± 0.007
Spambase	2/2	368/736 (20%)	368/736	919 (20%)	20/55	$\mathbf{0.885 \pm 0.01}$	0.883 ± 0.01	0.880 ± 0.03	0.838 ± 0.02
Magic	2/2	761/1522 (10%)	761/1522	3804 (20%)	28/79	$\mathbf{0.836 \pm 0.006}$	0.829 ± 0.006	0.827 ± 0.005	0.825 ± 0.005
Cod-rna	1/1	1325/1325 (1%)	1325/1325	66230 (20%)	37/37	$\mathbf{0.957 \pm 0.006}$	0.947 ± 0.008	0.951 ± 0.001	0.941 ± 0.006
Covertype	1/1	2760/2760 (1%)	2760/2760	29050 (5%)	53/53	$\mathbf{0.715 \pm 0.005}$	0.684 ± 0.008	0.697 ± 0.001	0.362 ± 0.003

Note: The reported (%) of the labeled data used in the learning process, is the percentage from $\mathcal{D} \backslash \mathcal{D}^{\text{test}}$, i.e. the test set is not included. The reported (%) of test set is the percentage from the entire data set.

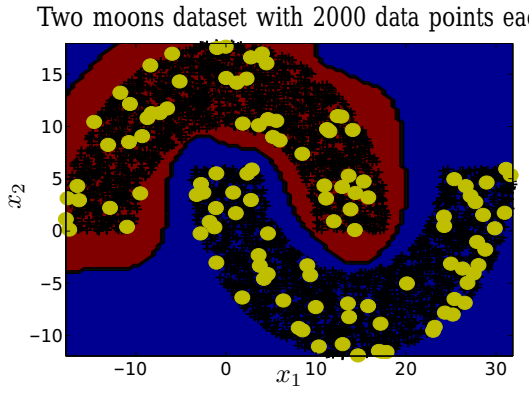

Fig. 1. The performance of the FS-MSS-KSC method with RBF kernel on two moons dataset yielding a sparse kernel-based model. In total there are 4000 data points. The prototype vectors (small working set) selected by the Rényi entropy criterion are depicted by circles.

apply an eigen-decomposition technique whereas RD-MSS-KSC does not apply any eigen-decomposition technique.

In Table IV, we examine the situation where the utilized size of unlabeled data is large and therefore applying LapSVMp will result in out-of-memory problem whereas the proposed FS-MSS-KSC and RD-MSS-KSC approaches that use an approximation of the feature map and reduced kernel matrix respectively, can deal with a large amount of unlabeled data points. Figure 3 shows the training computation times with respect to an increasing number of training points for Covertype data set. The RD-MSS-KSC showed a considerably reduced computation times due to the fact that, unlike FS-MSS-KSC, it does not involve an eigen-decomposition step.

VIII. CONCLUSIONS

In this paper, two approaches were proposed to make the semi-supervised KSC based algorithm scalable. The first approach uses the Nyström approximation of the feature map and solves the semi-supervised in the primal. The second approach solves the problem in the dual using a reduced kernel matrix. The first approach requires an eigen-decomposition technique to obtain the explicit feature map whereas the second one does not rely on any eigen-decomposition technique. The validity and applicability of the proposed methods is shown on real benchmark datasets. Both proposed approaches outperform the Laplacian SVM [4] in most cases in term of classification accuracy and training computation times. The training computational time taken by FS-MSS-KSC is longer than that of RD-MSS-KSC due to the involved eigen-decomposition step.

TABLE III

THE AVERAGE TRAINING COMPUTATION TIMES IN SECONDS FOR THE PROPOSED FIXED-SIZE, REDUCED MSS-KSC APPROACHES IN THIS PAPER, LAPSVMP [4] AND FIXED-SIZE LSSVM [11] METHODS ON REAL DATASETS OVER 10 SIMULATION RUNS.

Dataset	Training computation times in seconds			
	FS-MSS-KSC	**RD-MSS-KSC**	**LapSVMp**	**FS-LSSVM**
Heart	0.0090	0.0043	0.0267	0.0017
Pima-Indian	0.0381	0.0192	0.0295	0.0040
Spect	0.0081	0.0051	0.0265	0.0019
Iris	0.0090	0.0055	0.0025	0.0032
Ecoli	0.0395	0.0184	0.0030	0.0095
Satimage	0.1552	0.1192	0.2317	0.0277
Ring	0.0172	0.0139	0.1727	0.0069
Spambase	0.0246	0.0179	0.1497	0.0053
Magic	0.0737	0.0474	0.6026	0.0107
Cod-rna	0.3646	0.2349	7.6779	0.1590
Covertype	1.0721	0.7231	8.0201	0.6572

TABLE IV

THE AVERAGE TEST ACCURACY OF THE PROPOSED METHODS ON COVERTYPE DATASET. THE TEST SET IS 5% OF THE ENTIRE DATASET.

n_L/n_u	q_2/p	PV_L/PV_u	Method		
			FS-MSS-KSC	**RD-MSS-KSC**	**LapSVMp**
2760/2760	1/1	53/53	$\mathbf{0.715 \pm 0.01}$	0.684 ± 0.03	——–
2760/27600	0.5/10	53/84	$\mathbf{0.729 \pm 0.04}$	0.709 ± 0.05	——–
2760/55200	0.5/20	53/118	$\mathbf{0.731 \pm 0.02}$	0.712 ± 0.04	——–
2760/82800	0.5/30	53/144	$\mathbf{0.739 \pm 0.04}$	0.716 ± 0.03	——–
2760/138000	0.5/50	53/186	$\mathbf{0.742 \pm 0.05}$	0.723 ± 0.06	——–

ACKNOWLEDGMENTS

This work was supported by: • Research Council KUL: GOA/10/09 MaNet, PFV/10/002 (OPTEC), several PhD/postdoc & fellow grants • Flemish Government: ○ IOF: IOF/KP/SCORES4CHEM; ○ FWO: PhD/postdoc grants, projects: G.0377.12

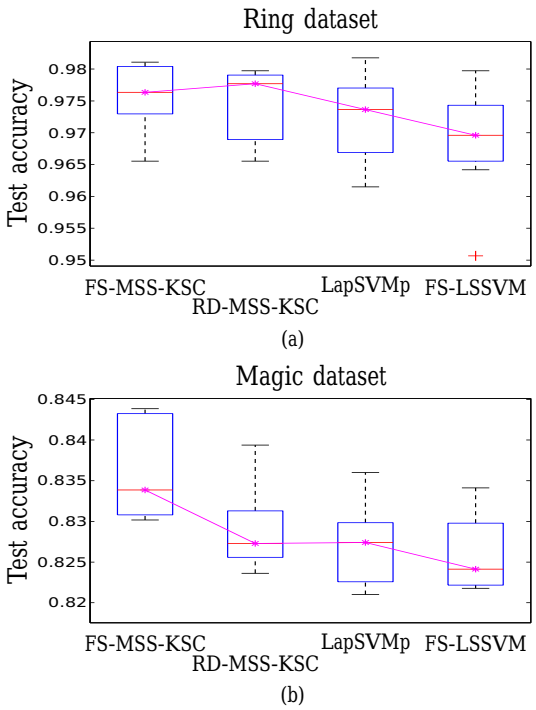

Fig. 2. Obtained test accuracy over 10 simulation runs using Fixed-size MSS-KSC, Reduced-MSS-KSC approaches proposed in this paper, LapSVMp [4] and Fixed-Size LSSVM [11] approaches for the two datasets (Ring and Magic) when RBF kernel is used.

(Structured systems), G.083014N (Block term decompositions), G.088114N (Tensor based data similarity); ○ IWT: PhD Grants, projects: SBO POM, EUROSTARS SMART; ○ iMinds 2013 ● Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 2012-2017) ● IBBT ● EU: FP7-SADCO (MC ITN-264735), ERC ST HIGHWIND (259 166), ERC AdG A-DATADRIVE-B (290923) ● COST: Action ICO806: IntelliCIS. Johan Suykens is a professor at the KU Leuven, Belgium.

REFERENCES

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

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

[3] 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.

[4] 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.

[5] 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.

[6] C. Alzate and J. A. K. Suykens, "Sparse kernel spectral clustering models for large-scale data analysis," *Neurocomputing*, vol. 74, no. 9, pp. 1382–1390, 2011.

[7] C. Alzate and J. A. K. Suykens, "A semi-supervised formulation to binary kernel spectral clustering," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2012, pp. 1992–1999.

[8] S. Mehrkanoon, C. Alzate, M. Raghvendra, R. Langone, and J. A. K. Suykens, "Multi-class semi-supervised learning based upon kernel spectral clustering," *Internal Report 13-146, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2013, submitted*, 2013.

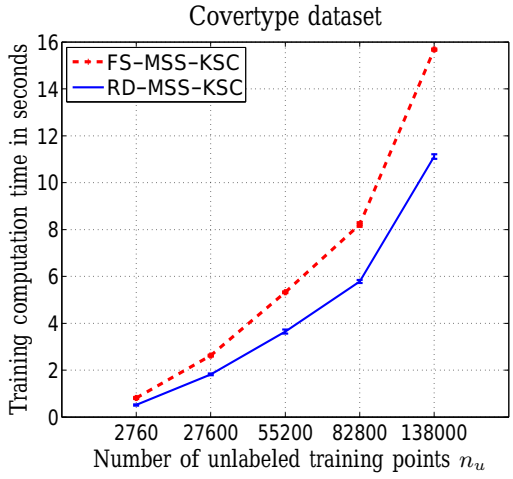

Fig. 3. Training computation time in seconds for the Covertype dataset with an increasing number of unlabeled training points and fix number of labeled points ($n_L = 2760$). The Reduced MSS-KSC approach takes less training time than the Fixed-Size MSS-KSC approach.

[9] C. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Advances in Neural Information Processing Systems 13*, 2001.

[10] C. T. Baker and C. Baker, *The numerical treatment of integral equations*. Clarendon press Oxford, 1977, vol. 13.

[11] 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.

[12] M. Espinoza, J. A. K. Suykens, and B. De Moor, "Fixed-size least squares support vector machines: A large scale application in electrical load forecasting," *Computational Management Science*, vol. 3, no. 2, pp. 113–129, 2006.

[13] 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.

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

[15] Y.-J. Lee and O. L. Mangasarian, "RSVM: Reduced support vector machines," in *Proceedings of the first SIAM international conference on data mining*. SIAM Philadelphia, 2001, pp. 5–7.

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

[17] 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.

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

[19] S. Mehrkanoon and J. A. K. Suykens, "Non-parallel semi-supervised classification based on kernel spectral clustering," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2012, pp. 2311–2318.

[20] J. C. Bezdek and N. R. Pal, "Some new indexes of cluster validity," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 28, no. 3, pp. 301–315, 1998.

[21] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[22] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002.

[23] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, no. 1, pp. 53–65, 1987.

[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.