Multi-Label Semi-Supervised Learning using Regularized Kernel Spectral Clustering

Siamak Mehrkanoon and Johan A.K. Suykens

Abstract— **Often in real-world applications such as web page categorization, automatic image annotations and protein function prediction, each instance is associated with multiple labels (categories) simultaneously. In addition, due to the labeling cost one usually deals with a large amount of unlabeled data while the fraction of labeled data points will typically be small. In this paper, we propose a multi-label semi-supervised kernel spectral clustering learning algorithm that learns from both labeled and unlabeled instances. The kernel spectral clustering algorithm (KSC) serves as a core model and the information of labeled data points is integrated into the model via regularization terms. The propagation of the multiple labels to unlabeled data points is achieved by incorporating the mutual correlation between (similarity across) labels as well as encouraging the model output to be as close as possible to the given ground-truth of the labeled data points. Thanks to the Nyström approximation method, an explicit feature map is constructed and the optimization problem is solved in the primal. Experimental results demonstrate the effectiveness of the proposed approaches on real multi-label datasets.**

I. Introduction

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. Furthermore, many real-world tasks are naturally posed as multi-label problems, where each data example may show multiple semantic meanings or concepts and consequently can be associated with multiple labels simultaneously. This is a generalized version of the most popular multi-class problems where each instance is restricted to have only one class label. Learning from multi-label data has recently received increased attention due to the ubiquitous presence of multi-label data in several application domains such as web page categorization, tag recommendation, gene function prediction, medical diagnosis and video indexing (see [1], [2], [3], [4]). Consider an example of multi-label classification in automatic image annotation task. An image can then be tagged as "Tree", "Building", "Car", "Street" and "Human" where each term represents a new semantic concept (see Fig. 1). Similarly, in text categorization a document can be assigned to multiple topics [5].

The typical solution of multi-label learning is to decompose the problem into a set of single-label problems [6]. The final labels of each instance are then obtained by using an aggregation scheme where the predictions of the individual classifiers are combined. Though this approach possesses the advantage of its simplicity, but as the correlation among labels are ignored, in certain situations it can show performance degradation. To exploit label correlations, different approaches have been proposed in the literature. The authors in [7] proposed an algorithm to extract the shared structures in multi-label dataset. [8] adopts a low rank structure to capture the complex correlations among labels and a graph based approach for learning from multi-label dataset is introduced in [9]. The authors in [10] proposed an approach which allows the label correlations to be exploited locally.





Fig. 1. The left figure can be tagged with: Ocean, Ship and Sky. The figure on the right can be tagged with: Human, Tree, Building, Car and Street.

Supervised multi-label learning problems have been extensively studied in the literature. However, in many real world applications, obtaining labeled data points is costly and time consuming and that becomes more prominent when one deals with a multi-label dataset. Semi-supervised learning is a framework in machine learning that aims at learning from both labeled and unlabeled data points [11]. Using unlabeled data together with labeled data often gives better results than using the labeled data alone.

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) [12], is one of the graph based methods which provide a natural out-of-sample extension. A transductive multi-label learning algorithm (TRAM) is proposed in [13] that aims at predicting the label sets of a group of unlabeled instances simultaneously using the information of both labeled and unlabeled data points. In [14], the authors proposed an approach based on constrained non-negative matrix factorization which is able to explore both the unlabeled data and the correlation among different classes simultaneously.

Kernel spectral clustering (KSC) is an unsupervised algo-

Siamak Mehrkanoon and Johan A.K. Suykens are with the Department of Electrical Engineering ESAT-STADIUS, KU Leuven, B-3001 Leuven, Belgium. (email: siamak.mehrkanoon@esat.kuleuven.be, mehrkanoon2011@gmail.com, johan.suykens@esat.kuleuven.be)

978-1-5090-0620-5/16/$31.00 ©2016 IEEE

rithm introduced in [15]. The primal problem of the kernel spectral clustering is formulated as a weighted kernel PCA. Recently Mehrkanoon *et al.* [16] proposed a multi-class semi-supervised algorithm (MSS-KSC) that focuses on single labeled classification and clustering. MSS-KSC is able to address both semi-supervised classification and clustering and it requires low dimensional embedding to reveal the underlying clusters for both static and non-stationary data streams [17]. A non-parallel semi-supervised classifier for binary single label dataset, where KSC acts as core model is also introduced in [18].

In this paper, we propose a new formulation by extending the MSS-KSC algorithm [16] to address the multi-label classification problem. KSC, an unsupervised algorithm, is used as core model and the available label information i.e. labeled data points as well as the underlying mutual correlation of the labels are incorporated into the model by means two regularization terms. The role of the regularization terms are to exploit the correlation among labels as well as enforcing the model output to be as close as possible the true set of labels provided by the user.

This paper is organized as follows. In Section II, a brief review of kernel spectral clustering is given. Section III, briefly reviews single label semi-supervised kernel spectral clustering. In Section IV, the multi-label semi-supervised kernel spectral clustering is formulated. Experimental results on real-life datasets are reported in Section V followed by concluding remarks in Section VI.

II. Brief overview of KSC

The kernel spectral clustering method is based on weighted kernel principal component analysis. It is described by a primal-dual formulation with 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 [15]:

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

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. Φ is the feature matrix defined as follows:

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

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. 1_n denotes a vector of all ones with size n. $V = \text{diag}(v_1, ..., v_n)$ with $v_i \in \mathbb{R}^+$ is a user defined weighting matrix. $w^{(\ell)}$ is the model parameters vector in the primal and are the bias terms.

Applying the Karush-Kuhn-Tucker (KKT) optimality conditions and eliminating the primal variables, one can obtain the solution in the dual 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 $VP_v\Omega$ has $k-1$ piecewise constant eigenvectors with eigenvalue 1. $b^{(\ell)}$ are the bias terms that result from the optimality conditions:

$$b^{(\ell)} = \frac{1}{1_n^T V 1_n} 1_n^T V \alpha^{(\ell)}, \ell = 1, \ldots, k-1.$$

The effect of centering matrix and bias terms is to center the kernel matrix Ω by removing the weighted mean from each column. Given this and due to the fact that the eigenvectors are piecewise constant, it is possible to use the eigenvectors corresponding to the first $k-1$ eigenvalues to partition the dataset into k clusters. The eigenvalue problem (2) is related to spectral clustering with random walk Laplacian when the weighting matrix V is taken as the inverse of a degree matrix. 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.

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 $\{x_i\}_{i=1}^{n_{\text{test}}}$ are 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. Single Label Semi-Supervised KSC

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:

$$Y_{ij} = \begin{cases} +1 & \text{if the } i\text{th point belongs to the } j\text{th class} \\ -1 & \text{otherwise.} \end{cases} \tag{3}$$

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) in primal is given as follows [16]:

$$\begin{aligned} \min_{w^{(\ell)},b^{(\ell)},e^{(\ell)}} \quad & \frac{1}{2}\sum_{\ell=1}^{Q} w^{(\ell)^T} w^{(\ell)} - \frac{\gamma_1}{2}\sum_{\ell=1}^{Q} e^{(\ell)^T} V e^{(\ell)} + \\ & \frac{\gamma_2}{2}\sum_{\ell=1}^{Q} (e^{(\ell)} - c^{(\ell)})^T A (e^{(\ell)} - c^{(\ell)}) \\ \text{subject to} \quad & e^{(\ell)} = \Phi w^{(\ell)} + b^{(\ell)} 1_n, \ \ell = 1, \ldots, Q, \end{aligned} \tag{4}$$

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

$$C = [c^{(1)}, \ldots, c^{(Q)}]_{n \times Q} = \left[\frac{0_{n_u \times Q}}{Y} \right]_{n \times Q}, \tag{5}$$

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:

$$A = \left[\begin{array}{c|c} 0_{n_u \times n_u} & 0_{n_u \times n_L} \\ \hline 0_{n_L \times n_u} & I_{n_L \times n_L} \end{array} \right], \tag{6}$$

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. The feature map φ can either be explicitly defined or implicitly by the kernel trick. It has been shown in [16] that the solution in the dual can be obtained by solving a linear system of equations of size n (number of data points). The score variables evaluated at the test set $\mathcal{D}^{\text{test}} = \{x_i\}_{i=1}^{n_{\text{test}}}$ become:

$$e_{\text{test}}^{(\ell)} = \Phi_{\text{test}} w^{(\ell)} + b^{(\ell)} 1_{n_{\text{test}}} \ \ell = 1, \ldots, Q, \tag{7}$$

where $\Phi_{\text{test}} = [\varphi(x_1), \ldots, \varphi(x_{n_{\text{test}}})]^T \in \mathbb{R}^{n_{\text{test}} \times m}$.

The decoding scheme for predicting the class membership of the test data points, 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 codebook $\mathcal{CB}$ 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 performance of the MSS-KSC on a synthetic problem is illustrated in Fig 2.

IV. Multi-Label Semi-Supervised KSC

In this section the Fixed-Size Multi-Label Semi-Supervised KSC (FS-MLSS-KSC) approach is formulated. Consider training data points

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

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 categories, then the label indicator matrix $Y \in \mathbb{R}^{n_L \times Q}$ is defined as previously in equation (3). Here as opposed to single label scenario, each instance can be assigned to multiple categories (see Table I). In our subsequent analysis, the i-th row and j-th column of the label indicator matrix Y will be referred to as Y_i and Y^j, respectively.

In order to extend the single label semi-supervised KSC model, described in the previous section, to address the multi-label learning problem, a new regularization term which aims at incorporating the correlation (similarity) among labels is added to the formulation (4). The similarity of the labels is calculated based on a kernel defined on the labels i.e. $K_{output}(Y^i, Y^j)$, where here Y^i and Y^j are the i-th and j-th columns of label indicator matrix Y for the multi-label dataset $\mathcal{D}$. The choices of the kernel will be discussed later. The new regularization term encourages the similarity of the score variables ($e^{(\ell)}$) of the model based on the similarity of the labels. In our multi-label learning formulation, an explicit feature map (see [19]) is used and the optimization problem is solved in the primal. In this way, not only the complexity of the algorithm is kept linear in the number of training data points, but also for this particular formulation (see section IV-B) we found it more straightforward to work with an explicit feature map.

TABLE I

Example of a multi-label dataset where each instance can be assigned to multiple-categories composed of both labeled and unlabeled instances.

	category_1	category_2	category_3	category_4
instance 1	?	?	?	?
instance 2	?	?	?	?
instance 3	?	?	?	?
instance 4	1	0	1	0
instance 5	1	1	1	0
instance 6	0	1	1	0
instance 7	0	1	0	1

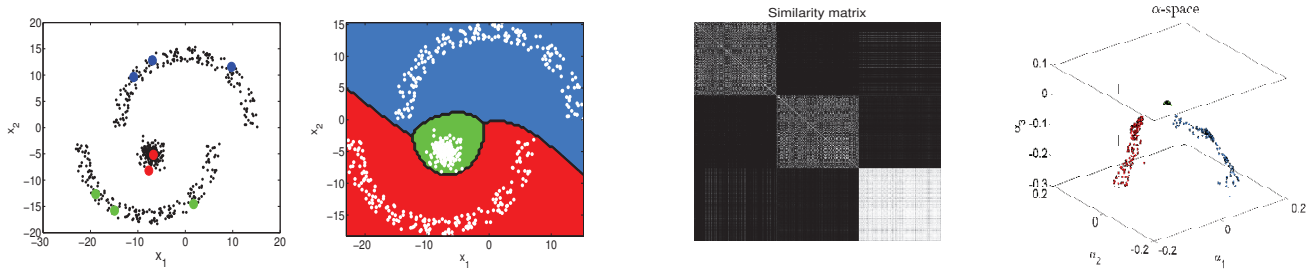
A. Explicit Feature Map

An explicit approximate expression for φ can be obtained by means of an eigenvalue decomposition of the kernel matrix Ω. When the size of the training dataset is large, the so called fixed-size approach [20], where the feature map is approximated by the Nyström method [21], [22], can be used. In what follows, we briefly summarize the fixed-size approach.

Consider the Fredholm integral equation of the first kind:

$$\int_C K(x, x_j)\phi_i(x)p(x)dx = \lambda_i \phi_i(x_j) \tag{8}$$

where C is a compact subset of $\mathbb{R}^d$. The approximation of the eigenfunctions $\phi_i(x)$ in (8) can be obtained by the Nyström method which applies a quadrature rule for discretizing the



(a) (b) (c) (d)

Fig. 2. Illustrating the performance of MSS-KSC model on synthetic single label example. (a) Original labeled and unlabeled data points. (b) The predicted memberships obtained using MSS-KSC model. (c) The associated similarity matrix indicating the cluster structure in the data. (d) The solution vector obtained by embedding the original data points to a new space (α-space).

left-hand side of (8). This will lead to the eigenvalue problem [21]:

$$\frac{1}{n}\sum_{k=1}^{n} K(x_k, x_j)u_{ik} = \lambda_i^{(s)}\, u_{ij} \tag{9}$$

where the eigenvalues λ_i and eigenfunctions ϕ_i from the continuous problem (8) 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 $\hat{\varphi} : \mathbb{R}^d \to \mathbb{R}^n$, for any point $x \in \mathbb{R}^d$, can be obtained as follows:

$$\hat{\varphi}_i(x) = \frac{1}{\lambda_i^{(s)}} \sum_{k=1}^{n} u_{ki}\, K(x_k, x) \tag{10}$$

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 $\hat{\varphi} : \mathbb{R}^d \to \mathbb{R}^m$ can be approximated as follows:

$$\hat{\varphi}(x) = [\hat{\varphi}_1(x), \ldots, \hat{\varphi}_m(x)]^T \tag{11}$$

where

$$\hat{\varphi}_i(x) = \frac{1}{\lambda_i^{(s)}} \sum_{k=1}^{m} u_{ki}\, K(x_k, x), i = 1, \ldots, m \tag{12}$$

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.

We aim at using an m-dimensional approximation to the feature map φ. Therefore we need to select a subset of fixed size m from a pool of training points of size n. As it has been motivated in [20], the Rényi entropy criterion [23] is used, to select m points from training dataset. Once the subset is available, the m-dimensional feature map is obtained using equation (12). It should be noted that m is a user defined parameter that can be designed in accordance with the available memory of the computer that is being used to conduct the experiments.

B. FS-MLSS-KSC formulation

Given the training dataset $\mathcal{D}$ and an m-dimensional approximation to the feature map, i.e.

$$\hat{\Phi} = [\hat{\varphi}(x_1), \ldots, \hat{\varphi}(x_n)]^T \in \mathbb{R}^{n\times m} \tag{13}$$

we formulate the Fixed-Size Multi-Label Semi-Supervised KSC (FS-MLSS-KSC) in primal as follows:

$$\begin{aligned} \min_{W} \quad & \frac{1}{2}Tr(W^TW) - \frac{\gamma_1}{2}Tr(E^TV_1E) \\ & - \frac{\gamma_2}{2}Tr(EV_2E^T) + \frac{\gamma_3}{2}Tr((E-C)^TA(E-C)) \\ \text{s.t} \quad & E = \tilde{\Phi}W, \end{aligned} \tag{14}$$

Here, $\tilde{\Phi} = [\hat{\Phi}, 1_n]$, where $\hat{\Phi}$ is an explicit feature provided by Nyström approximation method. $E = [e^{(1)}, \ldots, e^{(Q)}]_{n\times Q}$, $W = [w^{(1)}, \ldots, w^{(Q)}]_{(m+1)\times Q}$, C and A are defined as previously, see equations (5) and (6) respectively. $V_1 \in \mathbb{R}^{n\times n}$ and $V_2 \in \mathbb{R}^{Q\times Q}$ are the inverse of the input and output(category) degree matrix respectively defined as:

$$V_1 = \text{diag}(1/d_1^{in}, \cdots, 1/d_n^{in}),$$

and

$$V_2 = \text{diag}(1/d_1^{out}, \cdots, 1/d_Q^{out})$$

with $d_i^{in} = \sum_{j=1}^{n} K_{input}(x_i, x_j)$ and $d_i^{out} = \sum_{j=1}^{Q} K_{output}(Y^i, Y^j)$, where Y^i and Y^j are the i-th and j-th columns of label indicator matrix Y, respectively. The choices of the kernel function $K_{output}(\cdot,\cdot)$ are for instance linear kernel, normalized linear kernel, RBF kernel with the correlation distance [24] or cosine similarity. Here we use the cosine similarity [25] for the observed categories. The first two terms in the objective function of (14) correspond to the KSC core model which alone without any supervision is able to group data points that are similar to each other. In the case of well separated clusters and well tuned parameters, the score variables of data points that are similar in the input space form a line structure in the latent space, see [15] and [16]. The last two terms in the objective function of (14) are regularization terms which

aim at incorporating the similarity of the score variables as well as minimizing the difference between the model output and the true labels provided by the user. Specially, the role of the third term in the objective function of (14) is to incorporate the similarity at the category level to the model and encouraging the model to capture the similarity of the category space (clustering in the category space). The objective of the fourth regularization term is to enforce the score variables of the labeled instances to be as close as possible to the true underlying categories they belong. The second and third regularization terms act on the rows (instances) and columns (categories) of the score variable matrix E respectively. In this way, the correlation similarity of the labels is taken into account and FS-MLSS-KSC model learns the score variables E using the similarly of both instances as well as the similarity of the categories to which they belong.

Lemma 4.1: Given a finite dimensional (m-dimensional) approximation to the feature map $\hat{\Phi}$ and regularization constants $\gamma_1, \gamma_2, \gamma_3 \in \mathbb{R}^+$, the solution to (14) is obtained by solving the following linear system of equations equation:

$$\left[I_{(m+1)} + \tilde{\Phi}^T R \tilde{\Phi} - \frac{\gamma_2}{d_i^{out}} S\right] w^{(i)} = \gamma_3 \tilde{\Phi}^T c^{(i)}, i = 1, \ldots, Q \tag{15}$$

where $R = (\gamma_3 A - \gamma_1 V_1)$, $S = \tilde{\Phi}^T \tilde{\Phi}$ and $I_{(m+1)}$ is the identity matrix of size $(m+1) \times (m+1)$. Here $w^{(i)}$ and $c^{(i)}$ are the ith column of W and C, respectively.

Proof: Given the explicit feature map, one can rewrite (14) as an unconstrained optimization problem as follows,

$$\begin{aligned} \min_W \; J(W) = \frac{1}{2} Tr(W^T W) - \frac{\gamma_1}{2} Tr((\tilde{\Phi} W)^T V_1 (\tilde{\Phi} W)) - \\ \frac{\gamma_2}{2} Tr((\tilde{\Phi} W) V_2 (\tilde{\Phi} W)^T) \\ + \frac{\gamma_3}{2} Tr((\tilde{\Phi} W - C)^T A (\tilde{\Phi} W - C)). \end{aligned} \tag{16}$$

Taking the derivative of the cost function J with respect to W yields:

$$\frac{\partial \mathcal{J}}{\partial W} = 0 \Rightarrow$$

$$W + [\tilde{\Phi}^T (\gamma_3 A - \gamma_1 V_1) \tilde{\Phi}] W - \gamma_2 \tilde{\Phi}^T \tilde{\Phi} W V_2 = \gamma_3 \tilde{\Phi}^T C \tag{17}$$

and with some algebraic manipulation one can rewrite the above equation as in (15). ■

After obtaining the weight matrix W, the score variable matrix E for the test points $\mathcal{D}_{\text{test}} = \{x_1, \ldots, x_{n_{\text{test}}}\}$ can be computed as follows:

$$E_{\text{test}} = \tilde{\Phi}_{\text{test}} W, \tag{18}$$

where $\tilde{\Phi}_{\text{test}} = [\hat{\Phi}_{\text{test}}, 1_{n_{\text{test}}}]$ and $\hat{\Phi}_{\text{test}} = [\hat{\varphi}(x_1), \ldots, \hat{\varphi}(x_{n_{\text{test}}})]^T \in \mathbb{R}^{n_{\text{test}} m}$. The sign$(E_{\text{test}})$ can be used to predict the final labels for the test points. The procedure of the Fixed-Size Multi-label Semi-Supervised Kernel Spectral Clustering approach is summarized in Algorithm 1.

Algorithm 1: FS-MLSS-KSC approach for multi-label dataset.

Input: Training data set $\mathcal{D}$, multi labels indicator matrix Y, tuning parameters γ_1 and γ_2, γ_3 kernel parameters (if any), test set $\mathcal{D}^{\text{test}} = \{x_i\}_{i=1}^{n_{\text{test}}}$

Output: Label sets for test instances $\mathcal{D}^{\text{test}}$

1. Select m prototype vectors (small working set) using quadratic Rényi entropy criterion [23]. (see section IV. A)
2. Obtain the m-dimensional approximation of the feature map (13) by means of Nyström approximation (12).
3. Compute the weight matrix W using (15).
4. Estimate the test data projections E_{test} using (18).
5. Binarize the test projection to obtain the label set for test instances.

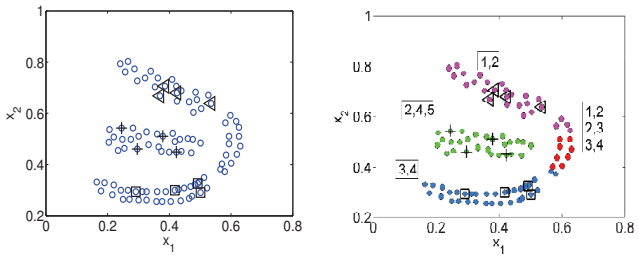
A Matlab demo of the algorithm can be downloaded at: `https://sites.google.com/site/smkmhr`

V. Numerical Experiments

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, γ_3 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. In this section experimental results on one synthetic as well as four real-life multi-label datasets from diverse domains are reported. The experiments are performed on a laptop computer with Intel Core i7 CPU and 8 GB RAM under Matlab 2014a.

From figure 3, one can observe that in total there are four clusters of data points with twelve labeled instances and a large number of unlabeled instances. In addition, one cluster does not have any labeled instances. Following Table 1, for this example, we have the following set of labels: {[1 1 0 0 0], [0 0 1 1 0], [0 1 0 1 1]} available. The proposed FS-MLSS-KSC approach is trained using both labeled and unlabeled instances and the obtained result is shown in Fig 3.b The predicted labels for the cluster that initially did not have any labeled instances, are {[0 1 1 0 0], [1 1 0 0 0], [0 0 1 1 0]}.

Descriptions of the used real-life datasets are summarized in Table II. We compare the performance of our proposed algorithm with several state-of-the-art multi-label classification algorithms, including TRAM [13], ML-kNN [28], ML-LOC [10]. TRAM is a transductive algorithm that uses both unlabeled and labeled data points in order to propagate the label set to unlabeled instances. Ml-kNN adapts the k-nearest



(a) (b)

Fig. 3. Illustrating the performance of MSS-KSC model on synthetic multi-abel example. (a) Synthetic multi-label data set where the set of labels are: {[1 1 0 0 0], [0 0 1 1 0], [0 1 0 1 1]}. (b) Labels predicted by FS-MLSS-KSC.

neighbors principle to multi-label datasets and often outperforms other multi-label algorithms. ML-LOC exploits the label correlation locally by enhancing the feature representation of each instance. For the competing algorithms, we use the parameter configuration suggested by the corresponding papers.

TABLE II
DATASET STATISTICS

Dataset	# Instances	# Attributes	# Labels
Yeast	1500	103	14
Natural Image	2000	294	5
Scene	2407	294	6
Emotion	593	72	6

In contrast with single label classification, in multi-label classification, predictions for an instance is a vector of labels and, therefore, the prediction can be fully correct, partially correct or fully incorrect. This makes evaluation of a multi-label classifier more challenging than that of a single label classifier. We evaluate the performance of the compared approaches using commonly used multi-label evaluation metrics: Hamming loss, ranking loss, Average and microF1 defined as follows (see [6] and references therein for more details). Let f be a multi-label classifier and with $Z_i = f(x_i)$ be the vector of label memberships predicted by f and Q be the finite set of class labels. Then the above-mentioned evaluation criteria which have been used in [6], [28], [13] are defined as follows:

- Micro F1 is defined as $\sum_{i=1}^{n} \frac{2|Y_i \cap Z_i|}{|Y_i|+|Z_i|}$. It evaluates both micro average of precision and micro average of recall with equal importance. The higher the value of Micro F1, the better the performance is.
- Hamming loss is defined as follows:

$$\frac{1}{nQ} \sum_{i=1}^{n} \sum_{j=1}^{Q} \Big[I(j \in Z_i \wedge j \notin Y_i) + I(j \notin Z_i \wedge j \in Y_i) \Big]$$

  where I is the indicator function. It evaluates how many times an instance-label pair is misclassified. The lower the value of Hamming loss, the better the performance is.
- Ranking loss is defined as follows:

$$\frac{1}{n} \sum_{i=1}^{n} \frac{1}{|Y_i||\bar{Y}_i|} |\{(y_1, y_2) \in Y_i \times \bar{Y}_i | h(x_i, y_1) < h(x_i, y_2)\}|$$

  where $\bar{Y}_i$ denotes the complementary set of Y_i in Q. It evaluates the average fraction of label pairs that are not correctly ordered. The lower the value of ranking loss, the better the performance is.
- Average precision, see [28], [13], evaluates the average fraction of labels ranked above a particular label $y \in Y_i$ which actually is in Y_i. The higher the value of average precision, the better the performance and average precision =1 means the perfect performance.

Four real-life datasets i.e. Yeast, Natural Scene Classification, Scene and Emotion are used in our experiments. In Yeast dataset the task is to predict the gene functional classes of the Yeast Saccharomyces cerevisiae [29]. The data set contains 1500 examples and 14 class labels. The Natural Scene dataset used in [28] consists of 2000 natural scene images belonging to the classes *desert*, *mountains*, *sea*, *sunset*, and *trees*. Over 22 images belong to multiple classes simultaneously and each image is associated with 1.24 class labels on average. The same method employed in [30] is used for extracting features from the given image and finally each image is described by a 294-dimensional feature vector. The multi-scene dataset is used in [30] and it consists of 2407 instances with 6 categories. The Emotions dataset has 72 attributes and 6 labels. The obtained results of the proposed FS-MLSS-KSC approach and those of the above mentioned algorithms are tabulated in Table III. In all the experiments, the given dataset is randomly partitioned to 70% training and 30% test sets respectively. The results on the test set are averaged over 10 simulation runs and reported in Table III.

In most of the cases studied here, the proposed FS-MLSS-KSC approach shows an improvement over the competitive algorithms with respect to four evaluation metrics. It is worth noting that the core model of FS-MLSS-KSC is an unsupervised algorithm that uses the labeled data points to guide the class memberships of the unlabeled instances.

We have also examined the scenario in which the number of labeled instances is changing. Figure 4, shows the performance of four algorithms, using four evaluation criteria, on the test set when different size of labeled training data is used. It can be observed that as the number of labeled training data points increases, the performance of all employed algorithms improves. Moreover, most of the time FS-MLSS-KSC shows a better performance in terms of Ranking Loss, Average precision, Hamming loss and MicroF1. Figure 5 illustrates the predicted labels for the given using the propose FS-MLSS-KSC approach.

VI. CONCLUSIONS

In this paper, a regularized kernel spectral clustering algorithm is proposed for classification of multi-label datasets. The model can learn from both labeled and unlabeled instances. The side information, i.e the labeled instances as

TABLE III

THE AVERAGE TEST ACCURACY OF THE PROPOSED FS-MLSS-KSC, TRAM [13], ML-LOC [10], AND ML-KNN [28] APPROACHES ON REAL DATASETS OVER 10 SIMULATION RUNS.

Dataset	n_L/n_u	$\mathcal{D}^{\text{test}}$	Evaluation Metric	**Method**			
				FS-MLSS-KSC	**TRAM**	**ML-LOC**	**ML-kNN**
Yeast	500/1000	917	Ranking Loss ↓	0.180 ± 0.004	0.181 ± 0.002	0.338 ± 0.007	0.182 ± 0.001
			Average precision ↑	0.751 ± 0.002	0.744 ± 0.001	0.712 ± 0.002	0.740 ± 0.002
			Hamming loss ↓	0.199 ± 0.003	0.218 ± 0.002	0.199 ± 0.001	0.207 ± 0.001
			MicroF1 ↑	0.626 ± 0.004	0.633 ± 0.003	0.619 ± 0.008	0.595 ± 0.004
Image	500/900	600	Ranking Loss ↓	0.173 ± 0.019	0.200 ± 0.009	0.171 ± 0.017	0.209 ± 0.019
			Average precision ↑	0.789 ± 0.019	0.757 ± 0.010	0.682 ± 0.026	0.750 ± 0.018
			Hamming loss ↓	0.168 ± 0.010	0.201 ± 0.007	0.176 ± 0.011	0.205 ± 0.007
			MicroF1 ↑	0.591 ± 0.025	0.553 ± 0.018	0.548 ± 0.046	0.350 ± 0.004
Scene	500/711	1196	Ranking Loss ↓	0.477 ± 0.031	0.514 ± 0.010	0.497 ± 0.015	0.522 ± 0.010
			Average precision ↑	0.576 ± 0.024	0.552 ± 0.019	0.516 ± 0.021	0.531 ± 0.031
			Hamming loss ↓	0.199 ± 0.027	0.213 ± 0.021	0.195 ± 0.014	0.221 ± 0.041
			MicroF1 ↑	0.420 ± 0.007	0.386 ± 0.009	0.403 ± 0.009	0.342 ± 0.005
Emotion	200/215	178	Ranking Loss ↓	0.205 ± 0.022	0.281 ± 0.015	0.333 ± 0.041	0.308 ± 0.023
			Average precision ↑	0.765 ± 0.015	0.689 ± 0.016	0.473 ± 0.048	0.670 ± 0.023
			Hamming loss ↓	0.222 ± 0.015	0.314 ± 0.009	0.299 ± 0.011	0.283 ± 0.009
			MicroF1 ↑	0.613 ± 0.037	0.500 ± 0.017	0.183 ± 0.046	0.296 ± 0.043

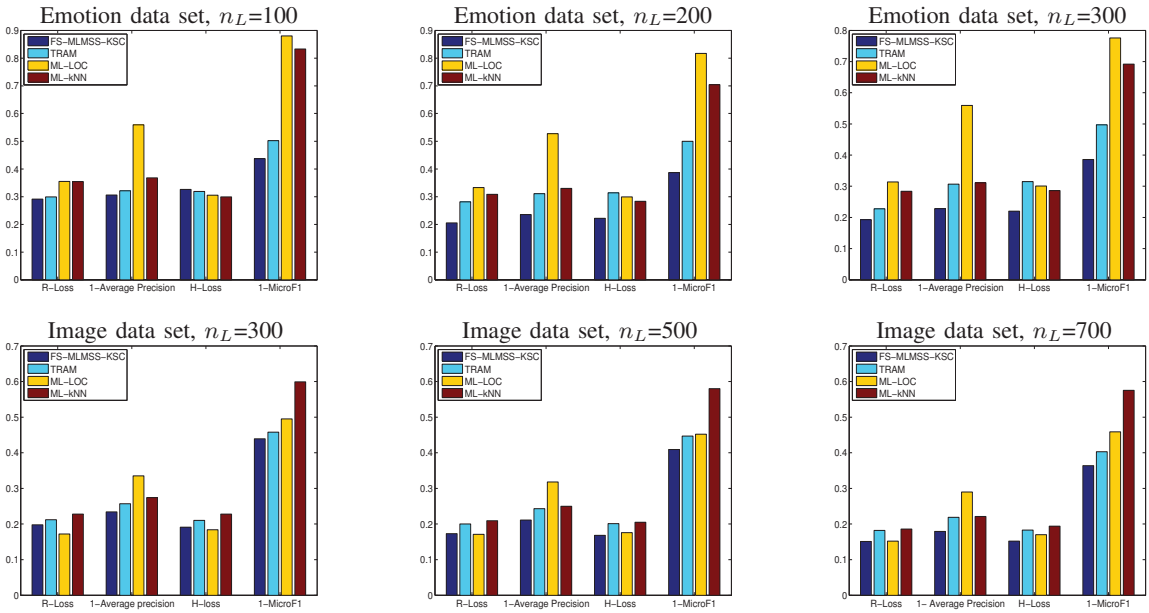


Fig. 4. The performance of the proposed model FS-MLSS-KSC as well as TRAM [13], ML-LOC [10], and ML-kNN [28] approaches on emotion and Image data sets when the number of available labeled data points (n_L) varies. Four different criteria are used for evaluating the performance of the employed models on the test data sets. As the number of labeled training data points increases, the performance with respect to the used evaluation metrics improves.



(a) FS-MLSS-KSC: *mountains, trees*. Ground truth: *mountains, trees*.

(b) FS-MLSS-KSC: *sea, sunset*. Ground truth: *sea, sunset*

(c) FS-MLSS-KSC: *desert, sunset*. Ground truth: *desert, sunset*

(d) FS-MLSS-KSC: *sea*. Ground truth: *mountains, sea*

Fig. 5. Examples of set of labels predicted by FS-MLSS-KSC on test instances from natural images scene data set [28].

well as their correlation, is incorporated to the KSC core model. The model uses an explicit feature map constructed by means of Nyström approximation method and the problem is solved in the primal making it potentially appealing for large scale multi-label problem. The validity and applicability of the proposed methods is shown on synthetic as well as real benchmark datasets.

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). Johan Suykens is a full professor at KU Leuven, Belgium.

REFERENCES

[1] L. Tang, S. Rajan, and V. K. Narayanan, "Large scale multi-label classification via metalabeler," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 211–220.

[2] G. Yu, C. Domeniconi, H. Rangwala, G. Zhang, and Z. Yu, "Transductive multi-label ensemble classification for protein function prediction," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 1077–1085.

[3] G.-P. Liu, G.-Z. Li, Y.-L. Wang, and Y.-Q. Wang, "Modelling of inquiry diagnosis for coronary heart disease in traditional chinese medicine by using multi-label learning," *BMC complementary and alternative medicine*, vol. 10, no. 1, p. 37, 2010.

[4] F. Markatopoulou, V. Mezaris, and I. Kompatsiaris, "A comparative study on the use of multi-label classification techniques for concept-based video indexing and annotation," in *MultiMedia Modeling*. Springer, 2014, pp. 1–12.

[5] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Multilabel text classification for automated tag suggestion," *ECML PKDD discovery challenge*, vol. 75, 2008.

[6] M. S. Sorower, "A literature survey on algorithms for multi-label learning," *Oregon State University, Corvallis*, 2010.

[7] S. Ji, L. Tang, S. Yu, and J. Ye, "Extracting shared subspace for multi-label classification," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 381–389.

[8] L. Xu, Z. Wang, Z. Shen, Y. Wang, and E. Chen, "Learning low-rank label correlations for multi-label classification with missing labels," in *Data Mining (ICDM), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1067–1072.

[9] Z.-J. Zha, T. Mei, J. Wang, Z. Wang, and X.-S. Hua, "Graph-based semi-supervised learning with multiple labels," *Journal of Visual Communication and Image Representation*, vol. 20, no. 2, pp. 97–103, 2009.

[10] S.-J. Huang, Z.-H. Zhou, and Z. Zhou, "Multi-label learning by exploiting label correlations locally." in *AAAI*, 2012.

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

[12] 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.

[13] X. Kong, M. K. Ng, and Z.-H. Zhou, "Transductive multilabel learning via label set propagation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 3, pp. 704–719, 2013.

[14] Y. Liu, R. Jin, and L. Yang, "Semi-supervised multi-label learning by constrained non-negative matrix factorization," in *Proceedings of the national conference on artificial intelligence*, vol. 21, no. 1. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, p. 421.

[15] 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.

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

[17] 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.

[18] S. Mehrkanoon and J. A. K. Suykens, "Non-parallel semi-supervised classification based on kernel spectral clustering," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2013, pp. 1–8.

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

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

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

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

[24] T. W. Liao, "Clustering of time series data–a survey," *Pattern recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.

[25] H. Wang, H. Huang, and C. Ding, "Image annotation using bi-relational graph of images and semantic labels," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 793–800.

[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.

[28] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[29] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Advances in neural information processing systems*, 2001, pp. 681–687.

[30] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.