Neurocomputing ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Neurocomputing

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

Sparse semi-supervised learning on low-rank kernel

Kai Zhang [a,*], Qiaojun Wang [b], Liang Lan [c], Yu Sun [d], Ivan Marsic [b]

[a] NEC Laboratories America, Inc., 4 Independence Way, Princeton, NJ, United States
[b] Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ, United States
[c] Huawei Noah's Ark Laboratory, Hong Kong
[d] Siemens Corporate Research, 755 College Road East, Princeton, NJ, United States

ARTICLE INFO

Article history:
Received 21 April 2013
Received in revised form
8 September 2013
Accepted 9 September 2013
Communicated by Shiliang Sun

Keywords:
Semi-supervised learning
Regularized least squares
Manifold regularization
Graph Laplacian
Sparse regression
Low-rank approximation

ABSTRACT

Advances of modern science and engineering lead to unprecedented amount of data for information processing. Of particular interest is the semi-supervised learning, where very few training samples are available among large volumes of unlabeled data. Graph-based algorithms using Laplacian regularization have achieved state-of-the-art performance, but can induce huge memory and computational costs. In this paper, we introduce L_1-norm penalization on the low-rank factorized kernel for efficient, globally optimal model selection in graph-based semi-supervised learning. An important novelty is that our formulation can be transformed to a standard LASSO regression. On one hand, this makes it possible to employ advanced sparse solvers to handle large scale problems; on the other hand, a globally optimal subset of basis can be chosen adaptively given desired strength of penalizing model complexity, in contrast to some current endeavors that pre-determine the basis without coupling it with the learning task. Our algorithm performs competitively with state-of-the-art algorithms on a variety of benchmark data sets. In particular, it is orders of magnitude faster than exact algorithms and achieves a good trade-off between accuracy and scalability.

1. Introduction

Advances of modern science and engineering in various domains have created unprecedented amount of data for information processing. Of particular interest is the semi-supervised learning scenario, where very few training labels are available due to the high cost of human interventions. How to utilize unlabeled data together with a small amount of labeled examples to boost learning performance while guaranteeing the algorithm efficiency has been a continued research interest. Enormous efforts have been devoted to semi-supervised learning, including transductive SVM [6,13], cotraining [3], label propagation [34], graph-based methods [1,20,36], semi-supervised kernel learning [4,7,15]. See a detailed survey in [35].

In this paper, we focus on a graph-based algorithm for semi-supervised learning. Assume that we use a positive semi-definite (PSD) kernel function $\kappa(\cdot,\cdot)$, and the $n\times n$ kernel/similarity matrix K such that $K_{ij}=\kappa(\mathbf{x}_i,\mathbf{x}_j)$. Define the graph Laplacian matrix as $\mathcal{L}=D-K$, where $D\in\mathbb{R}^{n\times n}$ is a (diagonal) degree matrix such that $D_{ii}=\sum_{j=1}^{n}K_{ij}$. The normalized graph Laplacian is defined as $\tilde{L}=\mathbf{I}-D^{-1/2}KD^{-1/2}$, where $\mathbf{I}$ is the identity matrix of proper size. The (normalized) graph Laplacian matrix imposes important smoothness constraints. To see this, suppose a prediction function $f(\cdot)$ is evaluated on $\{\mathbf{x}_i\}_{i=1}^{n}$, and the prediction is represented as $\mathbf{f}\in\mathbb{R}^{n\times 1}$ where $f_i=f(\mathbf{x}_i)$. Then the smoothness of $\mathbf{f}$ with regard to the graph is given by [1,37]

$$\sum_{i,j=1}^{n}\left(\frac{f_i}{\sqrt{D_{ii}}}-\frac{f_j}{\sqrt{D_{jj}}}\right)^2K_{ij}=\mathbf{f}'\tilde{L}\mathbf{f},$$

whose minimization is called the Laplacian regularization. It enforces a geometric, data-dependent constraint that the prediction should be sufficiently smooth with regard to the manifold structure of the data. Suppose that we are given a set of labeled data $\{\mathbf{x}_i\}_{i=1}^{l}$ and a large amount of unlabeled data $\{\mathbf{x}_i\}_{i=l+1}^{n}$, where $u=n-l$. By using a loss function $V(y,f(\mathbf{x}))$, Laplacian regularized semi-supervised learning can be formulated as [1]

$$\min_{f}\sum_{i=1}^{l}V(y_i,f(\mathbf{x}_i))+\gamma_A\|f\|_K^2+\gamma_I\frac{1}{n^2}\mathbf{f}'L\mathbf{f} \tag{1}$$

here $\|f\|_K$ is the Reproducing Kernel Hilbert Space (RKHS) norm of the prediction function, γ_A is the associated regularization parameter, and γ_I is the regularization parameter for the Laplacian smoothness term. The minimizer of this optimization problem admits the expansion form:

$$f^*(\mathbf{x})=\sum_{i=1}^{l+u}\boldsymbol{\alpha}_iK(\mathbf{x}_i,\mathbf{x}), \tag{2}$$

* Corresponding author.
E-mail address: kzhang@nec-labs.com (K. Zhang).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.09.033

where α_i's are the kernel expansion coefficients. Eq. (2) is called the representer theorem [1].

The Laplacian regularization has been proven as an effective way for semi-supervised learning [1,36]. One practical concern, however, is the need to manipulate the $n \times n$ kernel matrix which is the computational bottleneck. On the other hand, as the representer theorem (2) shows, the decision function is potentially spanned by all the labeled and unlabeled samples, leading to a dense model and slow testing.

Various attempts have been made to alleviate the computational cost of graph-based semi-supervised learning. One direction is to use low-rank approximation to scale up the optimization [10–12,26]. These algorithms are typically transductive, and the low-rank approximation does not consider label information which can be otherwise beneficial. Another direction is to span the model by only a small set of basis vectors [18,30], which will lead to fast training and testing. However, the selection of the basis is independent of the learning task. Also the training time scales quadratically with the model size, which is less efficient if a complex model is needed for difficult tasks.

Recently, the L_1-regularized linear regression, also known as the LASSO [24], has drawn considerable interest. It achieves simultaneous prediction and globally optimal model selection via penalizing the L_1-norm of the model coefficients. Inspired by it, we apply the L_1-norm penalization on the expansion coefficients of the low-rank factorized kernel in graph-based semi-supervised learning. To the best of our knowledge, applying the L_1-penalization with the low-rank kernel decomposition for semi-supervised learning is new. Our formulation not only ensures effective manifold regularization but also enjoys the globally optimal model selection. We also propose an efficient solution by approximately transforming our formulation to a standard LASSO, which is quite scalable to large data. Our algorithm competes favorably with exact, state-of-the-art algorithms such as Laplacian-RLS [1] and local and global consistency [34], while at the same time being orders of magnitudes faster. Compared with several fast semi-supervised learning algorithms [8,12,30], the accuracy of our algorithm is quite promising, though only a few times slower. Overall, our algorithm achieves a good trade-off between accuracy and scalability.

The rest of the paper is organized as follows. Section 2 introduces the proposed algorithm. In Section 3, we discuss related algorithms. Section 4 reports experimental results. The last section concludes the paper.

2. Proposed method

This section details our algorithm. First, we propose the mathematical formulation, i.e., L_1-penalization on low-rank kernel expansion in Laplacian-Regularized Least-Squares (Section 2.1). The resultant, sparse QP problem can be expensive to compute. So we propose to apply the Nyström low-rank approximation to the kernel matrix (Section 2.2), which allows us to transform the sparse QP to a standard LASSO (Section 2.3) that can be solved very efficiently.

2.1. L_1-penalization of Laplacian-regularized least squares

Given a set of training data $\{\mathbf{x}_i\}_{i=1}^{l}$ and unlabeled data $\{\mathbf{x}_i\}_{i=l+1}^{n}$, we can obtain kernel matrix K, degree matrix D, the graph Laplacian L and normalized graph Laplacian $\tilde{L}$ as in the previous section. For notational simplicity, we define $K_l \in \mathbb{R}^{l\times n}$ as the rows in the kernel matrix corresponding to the labeled samples. Note that this can also be written as $K_l = e_l K$ where $e_l = [\mathbf{I}_{l\times l}\ \mathbf{0}_{l\times u}]$.

By using (2), we assume that the classifier is spanned by all the labeled and unlabeled samples, i.e., $\mathbf{f} = K\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} \in \mathbb{R}^{n\times 1}$ is the kernel expansion coefficient.[1] We also require the model coefficients to be reasonably sparse considering the training and testing speed. Therefore we use an L_1-norm penalization on the model coefficients $\boldsymbol{\alpha}$ to control the model complexity. On the other hand, we require that the estimated labels, $K\alpha$, to be smooth with regard to the graph structure of the data similar to (1), and skipped the term $\|f\|_K$ for simplicity. Then we have the following problem:

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^{n\times 1}} \lambda_1 \|K_l\boldsymbol{\alpha} - \mathbf{y}_0\|^2 + (K\boldsymbol{\alpha})' L (K\boldsymbol{\alpha}) + \lambda_2 |\boldsymbol{\alpha}|_1 \tag{3}$$

here $\mathbf{y}_0 \in \mathbb{R}^{l\times 1}$ is the class labels for the labeled samples. The first term is a loss function that measures the discrepancy between the true and estimated labels on the labeled samples ($\mathbf{x}_l$). The second term enforces the smoothness constraint of $K\boldsymbol{\alpha}$. The third term $|\alpha|_1 = \sum_i |\alpha_i|$ is a regularization term, which encourages zero entries in the model coefficients α, thereby improving the efficiency of the testing phase.

The objective function (3) can be written equivalently as

$$\begin{aligned} &\min_{\boldsymbol{\alpha}\in\mathbb{R}^{n\times 1}} \quad \boldsymbol{\alpha}' Q \boldsymbol{\alpha} - 2c'\boldsymbol{\alpha} + \lambda_2 |\boldsymbol{\alpha}|_1 \\ &\text{where} \quad Q = K'LK + \lambda_1 K' e_l' e_l K \\ &c = K_l' \mathbf{y}_0 \end{aligned} \tag{4}$$

Formulation (4) is a quadratic programming problem with a sparsity constraint, which can be solved in different ways. For example, it can be re-formulated as a standard QP by decomposing α_i's as the difference of two non-negative terms a_i and b_i, which gives a standard QP with $2n$ variables with $2n$ non-negative constraints, and typically has a polynomial time complexity which is expensive for large n. Another possibility is to resort to an existing optimization technique like the alternating direction method [29]. This can provide exact optimal solution for (4).

In this paper, we are interested in obtaining an approximate, computationally more efficient solution of (4). An interesting observation here is that by fully exploiting the low-rank structure of the kernel matrix, problem (4) can be transformed to a standard LASSO regression, which can then be solved extremely efficiently due to the availability of various sparse solvers.

2.2. Low-rank approximation in semi-supervised setting

Low-rank matrix approximation is a useful tool for handling large matrices and reducing the dimensionality. Besides, it also has applications in dynamic systems [9,21]. In this paper, we are interested in the low-rank approximation of symmetric, positive semi-definite matrices (such as the kernel matrix) $K \in \mathbb{R}^{n\times n}$ in the form of

$$K \approx GG', \quad G \in \mathbb{R}^{n\times m}, \ m \ll n \tag{5}$$

here GG' is called the rank-m approximation of K. It has been found that in many learning problems, the kernel matrix typically has a fast decaying spectrum [27], justifying the use of the low-rank approximation technique in reducing the memory and computational cost. The optimal rank-m approximation is provided by the eigenvalue decomposition, which can be very expensive. So we will resort to a popular, sampling-based method called the Nyström method [11,10,26,31]. For an $n \times n$ kernel matrix, the Nyström method chooses a subset of m rows/columns $K_{nm} \in \mathbb{R}^{n\times m}$, compute an $m \times m$ eigenvalue decomposition on the intersection of selected rows and columns $K_{mm} \in \mathbb{R}^{m\times m}$, and then approximate

[1] In the case of multiple c classes, $\boldsymbol{\alpha} \in \mathbb{R}^{n\times c}$, which is a simple extension of the binary class formulation.

Please cite this article as: K. Zhang, et al., Sparse semi-supervised learning on low-rank kernel, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.09.033

the kernel matrix by

$$K \approx K_{nm} K_{mm}^{-1} K_{nm}' \tag{6}$$

Note that K_{nm} can be computed in $O(mn)$ time, and K_{mm}^{-1} in $O(m^3)$ time. Using (6), the kernel matrix can be written in factorized form as

$$K \approx GG', \quad G = K_{nm} K_{mm}^{-1/2} \tag{7}$$

in $O(m^2 n + m^3)$ time, which is linear since $m \ll n$. Recently, the Nyströ method has been used successfully to solve problems in semi-supervised matrix low-rank factorization [32] and transfer learning [33].

Next we consider how to obtain low-rank approximation of the Hessian matrix Q in (4), which will allow us to transform the sparse QP to a LASSO. However, the standard Nyström method is not fully suited here considering (i) the complex form of Q and (ii) the existence of labeled samples. Next we will revise the Nyström method and propose a new sampling scheme to approximate the Hessian in (4).

Note that the Hessian matrix (4) to be approximated can be written as

$$\begin{aligned} Q &= K'LK + \lambda_1 K' e_l' e_l K \\ &= K'(\mathbf{I} - D^{-1/2} K D^{-1/2} + \lambda_1 \Omega_l) K \\ &= K' D^{-1/2} (L + \lambda_1 D \Omega_l) D^{-1/2} K, \end{aligned}$$

where we define $\Omega_l = e_l' e_l$, a diagonal matrix with 1's on diagonal entries corresponding to labeled samples and 0's elsewhere. Note that the Hessian matrix has a multiplicative form:

$$\begin{aligned} &Q = PL_l P' \\ &\text{where} \quad P = K' D^{-1/2}, \ \ L_l = L + \lambda_1 D \Omega_l, \end{aligned} \tag{8}$$

with $L = D - K$ being graph the Laplacian matrix. In order to obtain the low-rank approximation of Q in the form of (5), both factors P and L_l need to be approximated in product form. To approximate P in factorization form, first we obtain approximation of the kernel matrix as in (6), then the degree matrix D can be approximated by

$$D \approx \mathrm{diag}(K_{nm} K_{mm}^{-1} K_{nm}' \mathbf{1}). \tag{9}$$

Therefore P (8) can be approximated in factorization form. However, it is nontrivial to obtain a decomposition of L_l (8) based on that of the kernel matrix K using closed-form operations, since L_l contains additive terms. The only way is to perform another low-rank approximation on L_l. However, the L_l incorporates both labeled and unlabeled information. Therefore it behaves differently from the kernel matrix and conventional sampling does not fit here. To see this, observe that Ω_l is a diagonal 0/1 matrix with non-zero diagonal entries only corresponding to labeled examples. Therefore, L_l is a *rectified Laplacian*, which augments the ith diagonal entry of L by $\lambda_1 D_{ii}$ if $\mathbf{x}_i$ is labeled. In such a case, the ith and jth diagonal entry of L_l would be significantly different even for two close samples $\mathbf{x}_i$ and $\mathbf{x}_j$ if one of them is label and the other is not. This indicates a strong non-smoothness (for a reasonably large λ_1) for which conventional sampling schemes would fail to accommodate.

Here we propose a unified sampling scheme that allows us to obtain low-rank approximation of Q with only one round of sampling, which can greatly improve the algorithm efficiency. Suppose we have l labeled samples, and are given a budget of selecting m landmark points, where we assume $l < m$. Then the landmark point set $\mathcal{Z}$ are chosen from two parts: (i) the labeled samples; (ii) a set of unlabeled points by the conventional sampling scheme. The former gives emphasis on label information, and the latter takes care of the overall data distribution, which can be selected using conventional sampling schemes on the unlabeled data. Suppose we have obtained m landmark points $\mathcal{Z}$, and approximate the kernel matrix K as in (6). Then both P and L_l (8) in the Hessian matrix Q can be approximated conveniently as follows:

$$\begin{aligned} &Q \approx (K_{nm} K_{mm}^{-1} K_{nm}' \tilde{D}) L_l (K_{nm} K_{mm}^{-1} K_{nm}' \tilde{D})' \\ &L_l = E W^{-1} E' \end{aligned} \tag{10}$$

where

$$\begin{aligned} &E = \tilde{D}^l(:, \mathcal{Z}) - K_{nm} \\ &W = \tilde{D}^l(\mathcal{Z}, \mathcal{Z}) - K_{mm} \\ &\tilde{D}_{ij}^l = \begin{cases} \tilde{D}_{ii} & \mathbf{x}_i \in \mathcal{Z}, \ \mathbf{x}_i \notin \mathcal{L}, \ i = j \\ \tilde{D}_{ii}(1 + \lambda_1) & \mathbf{x}_i \in \mathcal{L}, \ i = j \\ 0 & i \neq j \end{cases} \end{aligned} \tag{11}$$

With (10) and (11), the Hessian matrix can be decomposed into a low-rank factorization form as

$$\begin{aligned} &Q \approx FF' \\ &F = K_{nm} K_{mm}^{-1} K_{nm}' \tilde{D} E W^{-1/2}. \end{aligned} \tag{12}$$

2.3. LASSO: from quadratic function to least squares

In this section we show how the sparse quadratic programming problem (4) can be transformed to a least square problem. We have the following proposition.

Proposition 1. *Suppose the Hessian matrix Q in (4) is positive definite and has the eigenvalue decomposition*

$$Q = U \Lambda U', \tag{13}$$

then the sparse QP problem in (4) is equivalent to the following LASSO regression:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^{n \times 1}} \|A\boldsymbol{\alpha} - b\|_2^2 + \lambda_2 |\boldsymbol{\alpha}|_1$$

in that they have the same objective value given the same variable $\boldsymbol{\alpha}$. Here, ρ is a constant independent of any variables, and

$$A = \Lambda^{1/2} U', \quad b = \Lambda^{-1/2} U' c. \tag{14}$$

Proof. Compare the two expansions, $\boldsymbol{\alpha}' Q \boldsymbol{\alpha} - 2c'\boldsymbol{\alpha} + \rho$ and $\|A\boldsymbol{\alpha} - b\|_2^2 = \boldsymbol{\alpha}' A' A \boldsymbol{\alpha} - 2b' A \boldsymbol{\alpha} + b'b$, we have $Q = A'A$ and $c = A'b$. Suppose Λ is invertible,[2] by plugging the eigenvalue decomposition $Q = U \Lambda U'$ and the equality $UU' = U'U = \mathbf{I}$, we can show that $\boldsymbol{\alpha}' Q \boldsymbol{\alpha} - 2c'\boldsymbol{\alpha} + \rho = \|A\boldsymbol{\alpha} - b\|_2^2$ where $\rho = c' Q^{-1} c$ is a data-dependent constant. □

Proposition 1 shows that given the eigenvalue decomposition of the positive definite Hessian, a QP problem can be decomposed into complete least squares. As a result, the sparse problem (4) can be transformed equivalently to the LASSO regression problem.

In practice, we cannot obtain an exact eigenvalue decomposition due to the computational bottleneck, and the Hessian matrix typically contains many diminishing eigenvalues. Therefore, we relax the transform in Proposition 1 by replacing U and Λ (14) with U_m and Λ_m, the eigenvectors corresponding to the dominant m eigenvalues of Q.

Note that till now what we have about Q is a symmetric, low-rank decomposition (12), but not its eigenvectors as required in (13). Next we show how to compute the dominant eigenvectors of Q using (12) efficiently.

Proposition 2. *Given (12), the top m eigenvectors of Q can be approximated as follows. Compute the eigenvalue decomposition of*

[2] In case Λ is positive semi-definite, a pseudo-inverse or a small jittering factor (added to Λ) can be used.

Please cite this article as: K. Zhang, et al., Sparse semi-supervised learning on low-rank kernel, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.09.033

the $m \times m$ matrix $F'F = U_F \Lambda_F U_F'$. Then the eigenvalue and eigenvectors of Q can be approximated by $\tilde{\Lambda} = \Lambda_F$ and $\tilde{U} = F U_F \Lambda^{-1/2}$.

Proof. First we show that $\tilde{U}$ contains ortho-normal columns, such that $\tilde{U}'\tilde{U} = \Lambda^{-1/2} U_F'(F'F) U_F \Lambda^{-1/2} = \Lambda^{-1/2} U_F' U_F \Lambda_F U_F' U_F \Lambda^{-1/2} = \mathbf{I}$. Next we note that $\tilde{U}\tilde{\Lambda}\tilde{U}' = F U_F \Lambda^{-1/2} \Lambda \Lambda^{-1/2} U_F' F' = FF'$. These complete the proof. □

The whole algorithm is summarized in Algorithm 1. Note that steps 1–5 take $O(m^2 n + m^3)$ time. Available solvers for the LASSO regression in step 6 include [14,16]. These solvers have a very low complexity and as a result our algorithm is also very efficient and empirically has a linear time complexity with sample size.

Algorithm 1. Given l labeled samples, u unlabeled samples, landmark size m, output model coefficients $\boldsymbol{\alpha}$.

1: Select the landmark set $\mathcal{Z}$ by using labeled samples and $m-l$ unlabeled samples/cluster centers.
2: Compute K_{mm}, K_{nm} (5) and $\tilde{D}$ (9).
3: Obtain low-rank form of Q by (12), (10), and (11).
4: Apply Proposition 2 for approximate eigenvectors of Q.
5: Transform (4) to a standard LASSO by Proposition 1.
6: Use sparse solver to obtain optimal coefficients $\boldsymbol{\alpha}$.

3. Comparison with related methods

In [11], the Nyström method was used with Woodbury formula to scale up kernel matrix inverse in Gaussian process regression. Similarly, the idea was used in [34] for semi-supervised classification [12]. These algorithms are transductive and do not generalize to new samples. On the other hand, they use standard Nyström sampling without considering any labeled information. In comparison, our sampling scheme can accommodate the "discontinuities" induced by the labeled samples in the semi-supervised setting.

In the nonparametric function induction [8], the label of sample $\mathbf{x}_i$ is reconstructed using labels from a set of anchor points $\mathbf{x}_j$'s, i.e., $f(\mathbf{x}_i) = W_{ij} f(\mathbf{x}_j) / \sum_j W_{ij}$. Similar idea is used in the prototype vector machine [30], where two kinds of prototypes are employed for approximating the kernel matrix and the classifier, respectively. These two methods directly specify the kernel basis vectors independent of the learning task, and may therefore lead to suboptimal solutions. In comparison, our approach selects the basis vectors by coupling it explicitly in the discriminative (regression) setting, as a joint consequence of minimizing the loss, maximizing the smoothness, and maintaining a parsimonious model via L_1-regularization. Therefore our method provides a globally optimal model selection.

Another important difference is that in [8] or [30], the model size is directly determined by m, the number of landmark points. In case the learning task is difficult and a complex model is needed, these algorithms will have to choose a large number of landmark points to learn a complex model. Since complexity of these algorithms is quadratic with the m, doing this will be very expensive. In comparison, the model size in our algorithm does not directly hinge on m: m is only the number of landmark points needed to approximate the kernel matrix, and once the low-rank kernel matrix has been approximated, the model size will be determined adaptively by the L_1-norm regularization imposed on the low-rank factorized kernel. In other words, even very few landmark points are chosen, our approach can still obtain a dense model by using a relatively small regularization λ_2 (4), and, computing this model will be highly efficient due to its form of standard LASSO. In experiments we will further illustrate this advantage.

The idea of penalizing the L_1 norm of a kernel machine can be traced back to [2], which considers supervised setting and penalizes the L_1-norm of the dual variables in SVR with the ε-sensitive loss. The difficulty is that using a nonlinear kernel will lead to QP problem that is too expensive. Therefore a linear kernel is usually applied. A major contribution of our work is to transform the sparse QP problem using nonlinear kernels to a standard LASSO via low-rank approximation, which can then be solved quite efficiently.

In [25], the authors sparsify the Laplacian-SVM by an ε-sensitive loss function on the Laplacian regularization. In [22], a multi-view sparse SVM is pursued by using the ε-sensitive loss function with a globally optimal iterative algorithm. In [19], a sparse combination of the kernel eigenvectors is computed for prediction, which requires expensive eigenvalue decomposition. In [28,17], a sparse adjacency graph is built up by sparse regression of the sample coordinates and then used for semi-supervised learning. However this requires applying the LASSO regression repeatedly for each sample point and efficient alternative computing procedures have to be sought for large scale problems. In [23], a manifold-preserving sparse graph was proposed as a representation of sparsified manifolds, which is incorporated into the Laplacian support vector machine for semi-supervised learning problems. In comparison, we focus on the sparsity of the semi-supervised prediction function (in terms of the number of basis), but not the graph structure itself.

4. Experiments

4.1. An illustrative example

To examine the adaptive model selection of our method, we test it on the two-moon data, with only one labeled example from each class. We gradually increase the regularization parameter λ_2 and mark the selected basis in Fig. 1. As can be seen, when the regularization is strong, i.e., λ_2 is large, only samples close to labeled examples are chosen. As λ_2 decreases, more and more points along the actual decision boundary are included in the model. Another important observation is that the basis vectors chosen by our method do not directly depend on the landmark points chosen for the low-rank approximation of the kernel matrix.

4.2. Comparison with related algorithms

In this section, we use a number of benchmark data sets from the libsvm data sets[3] and semi-supervised learning book [5] to evaluate the performance of our algorithm, as summarized in Table 1. The following methods are compared: (1) LAP-RLS: Laplacian-regularized least-squares [1]; (2) LGC: learning with local and global consistency [34]; (3) NYS-LGC: standard Nyström low-rank approximation to scale up the local and global learning [12]; (4) NFI: nonparametric function induction[8]; (5) PVM: prototype vector machine [30]; (6) L1-LRK: our approach L_1-penalization on the low-rank kernel.

Our codes are written in Matlab and run on a Intel(R) Q9400 2.66 GHz PC with 4 GB memory. The experimental setting is as follows. The number of labels per each class is chosen as 50. The Gaussian kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 * \gamma)$ is used in our experiments. Parameters of different algorithms are chosen as follows. (1) LAP-RLS: the two regularization parameters in (1) are chosen as $\gamma_A \in \{10^{-2}, 1, 10^2\}$, $\gamma_I \in \{10^{-4}, 10^{-3}, \ldots, 10^2\}$; (2) LGC: the regularization parameter μ in Eq. (4) in [34] chosen from

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

Please cite this article as: K. Zhang, et al., Sparse semi-supervised learning on low-rank kernel, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.09.033

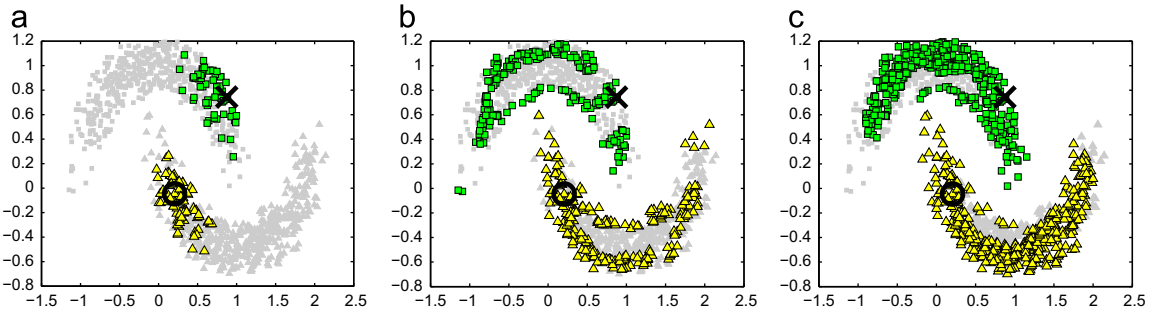


Fig. 1. When the L_1-regularization gradually diminishes, more and more basis vectors are included in the model, expanding from labeled samples to class boundaries, and ultimately to the whole data sets. (a) Large regularization, (b) median regularization and (c) small regularization.

Table 1
Summary of data sets.

Data	Sample size	Dimension	#classes	**Source**
G241C	1500	241	2	ssl book
G241D	1500	241	2	ssl book
DIGIT	1500	241	2	ssl book
USPS	1500	241	2	ssl book
COIL2	1500	241	2	ssl book
COIL	1500	241	6	ssl book
TEXT	1500	11960	2	ssl book
USPS56	1220	256	2	ssl book
USPS49	1296	256	2	ssl book
USPS38	1200	256	2	ssl book
USPS27	1367	256	2	ssl book
ADULT1-A	1605	123	2	ssl book
SEGMENT	2310	29	7	libsvm
SPLICE	3175	60	2	libsvm
DNA-FULL	3186	180	3	libsvm
SATIMAGE	6435	36	6	libsvm
SVMGUIDE-1A	7089	4	2	libsvm
USPS	7291	256	10	libsvm
MNIST	70 000	780	10	libsvm
REALSIM	72 309	20 958	2	libsvm
IJCNN	141 191	22	2	libsvm

$\{10^{-2}, 1, 10^2\}$; (3) NYS-LGC: similar to setting in LGC; (4) NFI: the regularization parameter for the loss term is chosen as $C_1 \in \{10^{-2}, 1, 10^2\}$; (5) PVM: the two regularization parameters in Eq. (9) are set as $C_1 \in \{10^{-2}, 1, 10^2\}$ and $C_2 = 0$; (6) L1-LRK: the two parameters in (4) are chosen as $\lambda_1 \in \{10^{-2}, 1, 10^2\}$, $\lambda_2 \in \{10^{-4}, 10^{-3}, \ldots, 10^2\}$. The number of landmark points m for methods (3)–(6) is chosen as $m = 10\%n$ if $n < 3000$; and $m = 200$ if $n > 3000$. The γ used in Gaussian kernel is chosen among $\gamma_0 \times 2^{\{-5,-4,-3,-2,-1,0,1,2,3,4,5\}}$ where γ_0 is the reciprocal of the averaged distance between data points. The reported results are based on the average of 30 random repeats.

The classification error and overall time consumption (training and testing) are summarized in Table 2. Since semi-supervised learning typically has very limited labels, it is impractical to use cross-validation for parameter selection. So we try different parameters and report the minimal error rate for each algorithm. As to the λ_2 in our method, since on many data sets a denser model tends to perform better, we report our results using a fixed regularization parameter $\lambda_2 = 0.1$, for which obtained models are sparse on most data sets.

As can be seen, our algorithm performs competitively with complete algorithms like Lap-RLS and LGC, but our algorithm is one or two orders of magnitude's faster and the larger the data size, the more obvious the efficiency. Compared with the fast algorithms like PVM, NYS-LGC, and NFI, the accuracy of our algorithm is quite promising, though the time consumption is typically several times slower. It can also be observed that on some data sets with relatively higher dimension, our approach performs very well. In practice, one can enforce stronger regularization for higher efficiency, say, let λ_2 grow with the number of samples. We perform paired Student-t test and for each task, we mark the winner(s) statistically better than the rest with a confidence level that is at least 95%. The number of winning tasks for each method is also listed in Table 2.

To further examine how the model sparsity affects the classification error, we gradually increase the L_1-regularization parameter λ_2 from 10^{-4} to 10^2. In Fig. 2, we plot the classification error with regard to the model size (m/n). As can be seen, on some data sets such as USPS and DIGIT (Fig. 2(b)), the classification accuracy is insensitive to the regularization parameter, meaning that our algorithm can almost always obtain a parsimonious model with accurate performance. On some data sets such as G241C and SPLICE (Fig. 2(a)), the classification accuracy deteriorates in the case of very sparse model, and gradually improves when the model becomes denser. This indicates the learning task is complicated and requires larger number of basis. On such data sets, there is still a flat domain of the error-vs-model curve, indicating that a reasonably sparse model can still be obtained with good predicting performance.

In Fig. 3(a), we examine the training time of PVM, NFI and our method with regard to the model size. For PVM and NFI, the model size is bounded by m; for our method, this is achieved by fixing m at a small value and changing the regularization parameter λ_2. As can be seen, as the intended model size increases, the training time of both PVM and NFI grows quickly; while that for our algorithm is much lower. This demonstrates the advantage of our algorithm in obtaining the complex model in an efficient manner. Fig. 3(b) plots the overall time consumption of our approach on the cover-type data with regard to (gradually increasing) sample size, which is very close to a linear trend.

In Fig. 4, we summarize the results in Table 2 by plotting the errors of the 6 algorithms on altogether 21 data sets. As can be seen, our algorithm achieves a good trade-off between scalability and accuracy.

5. Conclusion

In this paper, we propose to apply the L_1-norm penalization on the coefficients of a low-rank kernel expansion. The resultant problem can be transformed approximately to a standard LASSO, rendering great efficiency in seeking a globally optimal model in semi-supervised training. Competitive performance is obtained on a large number of benchmark data sets, in terms of the tradeoff between scalability and accuracy in semi-supervised classification problem. Our work can be

Please cite this article as: K. Zhang, et al., Sparse semi-supervised learning on low-rank kernel, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.09.033

Table 2
The classification error (%) and time consumption (in seconds) of different algorithms.

Data	Exact methods		Approximate methods			
	LGC (transductive)	LAP-RLS (inductive)	NYS-LGC (transductive)	NFI (inductive)	PVM (inductive)	L1-LRK (inductive)
Error						
G241C	23.03 ± 2.09	24.82 ± 3.50	23.97 ± 2.73	28.40 ± 7.01	27.85 ± 2.78	**17.98 ± 1.41**
G241D	25.56 ± 3.85	25.21 ± 3.32	29.33 ± 4.60	29.14 ± 5.57	24.49 ± 2.28	**22.41 ± 3.05**
DIGIT	**3.19 ± 0.72**	4.93 ± 1.07	5.46 ± 1.08	4.62 ± 1.05	3.83 ± 0.80	4.82 ± 1.07
USPS	**4.42 ± 0.70**	7.85 ± 1.56	7.62 ± 1.39	5.15 ± 0.76	5.85 ± 0.76	7.28 ± 1.04
COIL2	**10.06 ± 2.95**	11.87 ± 1.92	15.86 ± 2.39	12.63 ± 2.60	12.52 ± 2.27	10.72 ± 1.90
COIL	**6.41 ± 0.91**	10.17 ± 1.37	17.99 ± 1.92	56.39 ± 1.07	11.74 ± 1.63	10.27 ± 1.37
TEXT	**22.30 ± 0.59**	24.47 ± 0.87	24.95 ± 2.55	30.53 ± 1.64	28.89 ± 2.33	23.49 ± 2.14
USPS56	3.11 ± 0.73	2.39 ± 0.62	4.43 ± 1.34	2.97 ± 0.60	3.28 ± 0.90	**1.99 ± 0.43**
USPS49	7.83 ± 1.74	**3.46 ± 1.29**	9.91 ± 2.09	7.46 ± 1.35	7.54 ± 1.17	3.78 ± 1.16
USPS38	3.85 ± 0.66	**3.78 ± 0.79**	5.25 ± 0.69	**3.85 ± 0.77**	4.74 ± 0.73	**3.68 ± 0.64**
USPS27	2.99 ± 0.71	**1.43 ± 0.25**	3.14 ± 0.56	3.59 ± 0.89	2.14 ± 0.71	2.29 ± 0.61
ADULT1-A	23.01 ± 0.27	29.77 ± 2.18	**22.35 ± 1.28**	28.76 ± 1.91	**22.38 ± 1.06**	25.42 ± 1.71
SEGMENT	7.68 ± 0.95	7.35 ± 0.83	9.28 ± 1.10	9.18 ± 0.89	**7.16 ± 0.68**	**7.25 ± 0.86**
SPLICE	27.95 ± 1.89	29.12 ± 4.42	29.73 ± 3.63	28.75 ± 2.73	26.09 ± 1.86	**22.80 ± 3.13**
DNA	27.08 ± 3.95	24.13 ± 2.98	31.02 ± 3.10	42.67 ± 3.10	20.83 ± 1.31	**19.48 ± 1.85**
SATIMAGE	**14.46 ± 0.41**	14.54 ± 0.57	16.00 ± 0.41	23.54 ± 0.78	14.64 ± 0.70	14.57 ± 0.63
SVMGUIDE-1A	4.73 ± 0.53	6.17 ± 0.60	**4.69 ± 0.34**	4.78 ± 0.54	4.82 ± 0.47	**4.58 ± 0.44**
USPS	**6.63 ± 0.57**	8.57 ± 0.35	18.67 ± 1.38	13.97 ± 1.34	7.35 ± 0.64	8.78 ± 0.30
MNIST	–	–	20.27 ± 0.11	13.96 ± 0.23	**11.08 ± 0.13**	13.20 ± 0.07
REAL-SIM	–	–	13.01 ± 0.89	**12.86 ± 1.08**	20.2 ± 3.19	**12.82 ± 0.84**
IJCNN	–	–	9.59 ± 0.05	15.41 ± 0.98	**9.20 ± 0.52**	9.59 ± 0
# Wins	7	3	2	2	4	9
Overall time						
G241C	58.45	72.60	0.25	0.13	0.63	0.64
G241D	76.42	57.18	0.25	0.14	0.65	0.66
DIGIT	67.89	56.64	0.24	0.12	0.63	0.55
USPS	42.60	40.74	0.20	0.11	0.64	0.49
COIL2	67.60	44.91	0.15	0.08	0.45	0.62
COIL	57.11	38.31	0.14	0.07	0.43	0.81
TEXT	112.25	133.75	0.83	1.48	11.93	15.42
USPS56	30.71	28.94	0.70	0.20	0.98	0.55
USPS49	19.55	23.03	0.73	0.28	1.27	0.57
USPS38	15.65	11.60	0.28	0.20	0.73	0.31
USPS27	34.12	35.43	0.52	0.27	1.12	0.66
ADULT1-A	49.74	37.15	0.48	0.16	0.71	0.71
SEGMENT	190.22	110.00	0.46	0.19	0.76	6.06
SPLICE	606.33	596.22	0.55	0.19	1.03	1.16
DNA	619.75	605.90	0.30	0.15	0.96	2.80
SATIMAGE	3203,46	3259.88	0.43	0.23	1.28	6.56
SVMGUIDE-1A	4180.50	4082.17	0.48	0.32	1.26	13.74
USPS	5182.86	4582.87	0.48	0.37	2.39	15.85
MNIST	–	–	137.11	117.15	917.67	215.52
REAL-SIM	–	–	22.90	20.97	231.93	38.61
IJCNN	–	–	51.35	48.33	267.08	53.81

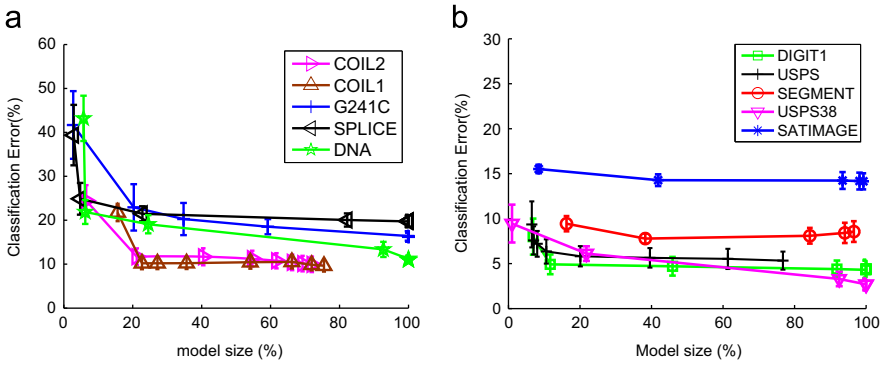


Fig. 2. Classification error versus the L_1-norm regularization parameter λ_2. (a) Gradually decreasing error rate w.r.t. λ_2 and (b) stable error rate w.r.t. λ_2.

extended in several directions. First, we will conduct more principled analysis on how the low-rank approximation affects the optimal solution of the sparse quadratic programming problem. Second, we will expand our algorithm to the use of hinge loss function in SVM. Third, we can apply adaptive, data-dependent sparsity constraint [38], which is expected to provide better results.

Please cite this article as: K. Zhang, et al., Sparse semi-supervised learning on low-rank kernel, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.09.033

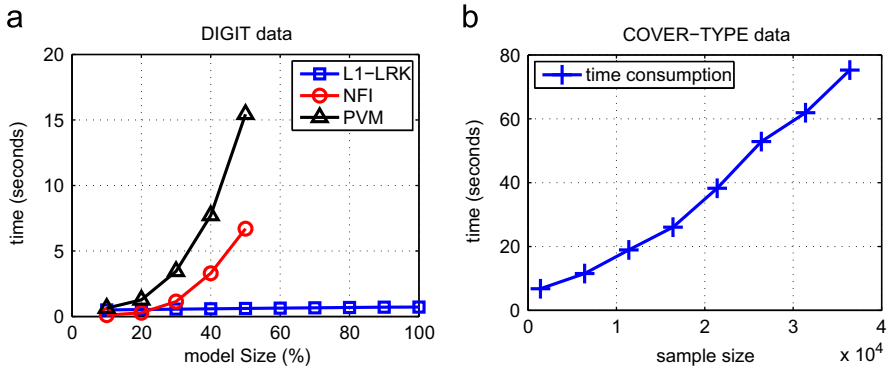


Fig. 3. Training (a) and overall time consumption (b) of our approach. (a) Training time versus different model sizes for PVM, NFI, and our approach and (b) the overall time consumption (training and testing) of our algorithm versus the sample size.

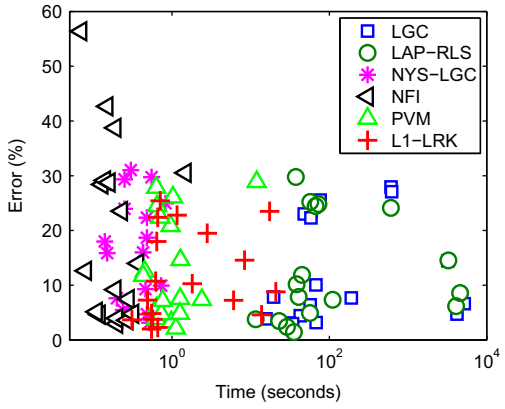


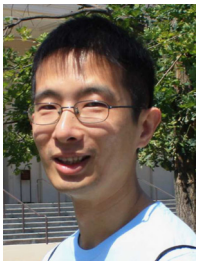
Fig. 4. Error versus time for all 6 methods on 21 data sets.

References

[1] M. Belkin, P. Niyogi, Y. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. (2006) 2399–2434.
[2] J. Bi, K.P. Bennett, M. Embrechts, C.M. Breneman, M. Song, I. Guyon, A. Elisseeff, Dimensionality reduction via sparse support vector machines, J. Mach. Learn. Res. 3 (2003) 1229–1243.
[3] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Annual Conference on Computational Learning Theory, 1998, pp. 92–100.
[4] O. Chapelle, J. Weston, B. Schölkopf, Cluster kernels for semi-supervised learning, in: Neural Information Processing Systems, 2003, pp. 585–592.
[5] O. Chapelle, B. Schölkopf, A. Zien (Eds.), Semi-Supervised Learning, MIT Press, 2006.
[6] R. Collobert, F. Sinz, J. Weston, L. Bottou, T. Joachims, Large scale transductive SVMs, J. Mach. Learn. Res. (2006) 1687–1712.
[7] C. Cortes, M. Mohri, A. Rostamizadeh, Two-stage learning kernel algorithms, in: International Conference on Machine Learning, 2010.
[8] O. Delalleau, Y. Bengio, N. Roux, Efficient non-parametric function induction in semisupervised learning, in: International Workshop on Artificial Intelligence and Statistics, 2005, pp. 96–103.
[9] K. Deng, Y. Sun, P.G. Mehta, S.P. Meyn, An information-theoretic framework to aggregate a Markov chain, in: American Control Conference, 2009, pp 731–736.
[10] P. Drineas, M.W. Mahoney, On the Nyström method for approximating a gram matrix for improved kernel-based learning, J. Mach. Learn. Res. 6 (2005) 2153–2175.
[11] C. Fowlkes, S. Belongie, F. Chung, J. Malik, Spectral grouping using the Nystrom method, IEEE Trans. Pattern Anal. Mach. Intell. (2004) 214–225.
[12] C. Gustavo, T. Marsheva, D. Zhou, Semisupervised graph-based hyperspectral image classification, IEEE Trans. Geosci. Remote Sensing (2007) 3044–3054.
[13] T. Joachims, Transductive inference for text classification using support vector machines, in: International Conference on Machine Learning, 1999, pp. 200–209.
[14] S. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinevsky, An interior-point method for large-scale l1-regularized least squares, IEEE J. Sel. Top. Signal Process. 1 (4) (2007) 606–617.
[15] G. Lanckriet, N. Cristianini, P. Bartlett, L.E. Ghaoui, M. Jordan, Learning the kernel matrix with semidefinite programming, J. Mach. Learn. Res. (2004) 27–72.
[16] J. Liu, S. Ji, J. Ye, SLEP: Sparse Learning with Efficient Projections, Arizona State University, 2009.
[17] W. Liu, J. He, S. Chang, Large graph construction for scalable semi-supervised learning, in: International Conference on Machine Learning, 2010.
[18] M. Mohri, A. Talwalkar, Sampling techniques for the Nyström method, J. Mach. Learn. Res. (2009) 304–311.
[19] K. Sinha, M. Belkin, Semi-supervised learning using sparse eigenfunction, in: Neural Information Processing Systems, 2009, pp. 1687–1695.
[20] A. Smola, R. Kondor, Kernels and regularization on graphs, in: Annual Conference on Learning Theory, 2003, pp. 144–158.
[21] Y. Sun, P.G. Metha, The Kullback–Leibler rate pseudo-metric for comparing dynamical systems, IEEE Trans. Autom. Control 55 (July (7)) (2010) 1585–1598.
[22] S. Sun, J. Shawe-Taylor, Sparse semi-supervised learning using conjugate functions, J. Mach. Learn. Res. 11 (2010) 2423–2455.
[23] S. Sun, Z. Hussain, J. Shawe-Taylor, Manifold-preserving graph reduction for sparse semi-supervised learning, Neurocomputing 124 (26) (2014) 13–21.
[24] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B 58 (1996) 267–288.
[25] I.W. Tsang, J.T. Kwok, Large-scale sparsified manifold regularization, Neural Inf. Process. Syst. 19 (2006) 1401–1408.
[26] C. Williams, M. Seeger, Using the Nyström method to speed up kernel machines, in: Neural Information Processing Systems, 2001, pp. 682–688.
[27] C. Williams, M. Seeger, The effect of the input density distribution on kernel-based classifiers, in: Proceedings of the 17th International Conference on Machine Learning, 2000, pp. 1159–1166.
[28] S. Yan, H. Wang, Semi-supervised learning by sparse representation, in: SIAM Conference on Data Mining, 2009, pp. 792–801.
[29] J. Yang, Y. Zhang, Alternating direction algorithms for l1 problems in compressive sensing, SIAM J. Sci. Comput. 33 (1) (2011) 250–278.
[30] K. Zhang, J.T. Kwok, Prototype vector machine for large scale semi-supervised learning, in: International Conference on Machine Learning, 2009, pp. 1233–1240.
[31] K. Zhang, I. Tsang, J.T. Kwok, Improved Nyström low-rank approximation and error analysis, in: International Conference on Machine Learning, 2008, pp. 1232–1239.
[32] K. Zhang, L. Lan, J. Liu, A. Rauber, F. Moerchen, Inductive kernel low-rank decomposition with priors, in: International Conference on Machine Learning, 2012, pp. 305–312.
[33] K. Zhang, W. Zheng, Q. Wang, J.T. Kwok, Q. Yang, I. Marsic, Covariate shift in Hilbert space: a surrogate kernel approach, in: International Conference on Machine Learning, 2013, pp. 388–395.
[34] C. Zhou, O. Bousquet, T.N. Lal, J. Weston, B. Schölkopf, Learning with local and global consistency, Neural Inf. Process. Syst. 16 (2004) 321–328.
[35] X. Zhu. Semi-Supervised Learning Literature Survey, Technical Report, Department of Computer Science, University of Wisconsin-Madison, 2006.
[36] X. Zhu, Z. Ghahramani, J. Lafferty, Semi-supervised learning using gaussian fields and harmonic functions, in: International Conference on Machine Learning, 2003, pp. 912–919.
[37] X. Zhu, J. Kandola, Z. Ghahbramani, J. Lafferty, Nonparametric transforms of graph kernels for semi-supervised learning, Neural Inf. Process. Syst. 17 (2005) 1641–1648.
[38] H. Zou, The adaptive LASSO and its oracle properties, J. Am. Stat. Assoc. 101 (476) (2006) 1418–1429.

Please cite this article as: K. Zhang, et al., Sparse semi-supervised learning on low-rank kernel, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.09.033

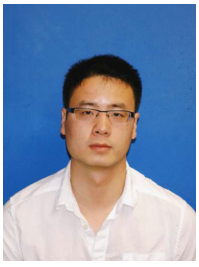
Kai Zhang received the Ph.D. degree in computer science from the Hong Kong University of Science and Technology in 2008. Then he joined the Life Sciences Division, Lawrence Berkeley National Laboratory in Berkeley, CA, as a postdoc researcher. He is currently with NEC Laboratories America, Inc. in Princeton, NJ. His research interests include large scale machine learning, bioinformatics and complex networks.



Yu Sun received the B.S. degree in Automation from the University of Science and Technology of China in 2004. He received the Master of Engineering degree in Electrical Engineering from University of Notre Dame in 2006, and Ph.D degree in Mechanical Engineering from University of Illinois, Urbana-Champaign in 2011. His research interests include estimation theory, machine learning, fault diagnosis, and stochastic dynamical systems.

Qiaojun Wang is a Ph.D. student in the Department of Electrical and Computer Engineering at Rutgers University. He received a B.E. (2007) from Shandong Polytechnic University, China, and MS (2009) from Stevens Institute of Technology, NJ. His research interests include semi-supervised learning, transfer learning and feature selection.

Ivan Marsic received the Dipl.Ing. and M.S. degrees in computer engineering from the University of Zagreb, Croatia, and the Ph.D. degree in biomedical engineering from Rutgers University, New Brunswick, NJ, USA, in 1994. He is currently a Professor with the Department of Electrical and Computer Engineering, Rutgers University. His current research interests include sensor networks and machine learning for healthcare applications.



Liang Lan received the B.S. degree in bioinformatics from Huazhong University of Science and Technology, Wuhan, China, in 2007, and the Ph.D. degree in computer and information sciences from Temple University, Philadelphia, PA in 2012. He is currently a researcher of Huawei Noah's Ark Lab. His research interests are data mining and machine learning.

Please cite this article as: K. Zhang, et al., Sparse semi-supervised learning on low-rank kernel, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.09.033