Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/080435

Semi-Supervised Learning by Local Behavioral Searching Strategy

Chun Zhang[1], *Junan Yang*[1,*], *Jiyang Zhang*[2], *Dongsheng Li*[1] *and Aixia Yong*[1]

[1] Electronic Engineering Institute, Key Laboratory of Electronic Restriction, 230037 Hefei, China
[2] Jiangnan Institute of Computing Technology, 214083 Wuxi, China

Received: 5 Aug. 2013, Revised: 7 Nov. 2013, Accepted: 8 Nov. 2013
Published online: 1 Jul. 2014

Abstract: Semi-supervised learning has attracted a significant amount of attention in pattern recognition and machine learning. Among these methods, a very popular type is semi-supervised support vector machines. However, parameter selection in heat kernel function during the learning process is troublesome and harms the performance improvement of the hypothesis. To solve this problem, a novel local behavioral searching strategy is proposed for semi-supervised learning in this paper. In detail, based on human behavioral learning theory, the support vector machine is regularized with the un-normalized graph Laplacian. After building local distribution of feature space, local behavioral paradigm considers the form of the underlying probability distribution in the neighborhood of a point. Validation of the proposed method is performed with extensive experiments. Results demonstrate that compared with traditional method, our method can more effectively and stably enhance the learning performance.

Keywords: semi-supervised learning, behavioral learning, regularization, manifold learning, support vector machines

1 Introduction

Semi-supervised learning has received a significant interest in pattern recognition and machine learning. It exploits unlabeled data in addition to the limited labeled ones to improve the learning performance [1]. Many semi-supervised learning algorithms have been proposed during the past decade, among which a very popular type of algorithms is the semi-supervised support vector machines (S3VMs).

Examples of this type include the semi-supervised SVM [2], the transductive SVM (TSVM) [3], and the Laplacian SVM [4]. S3VM and the TSVM are built upon the cluster assumption and use the unlabeled data to regularize the decision boundary. Specifically, these methods prefer the decision boundary to pass through low-density regions [5]. The Laplacian SVM is a S3VM that exploits the datas manifold structure via the graph Laplacian. It encodes both the labeled and unlabeled data by a connected graph, where each instance is represented as a vertex and two vertices are connected by an edge if they have large similarity. The goal is to find class labels for the unlabeled data such that their inconsistencies with both the supervised data and the underlying graph structure are minimized.

However, while many efficient SVM methods have been developed for supervised learning, S3VMs still suffer from inefficiency issues. In particular, the optimization problem of Bennett and Demirizs S3VM is formulated as a mixed-integer programming problem and so is computationally intractable in general. TSVM, on the other hand, iteratively solves standard supervised SVM problems. However, the number of iterations required may be large since the TSVM is based on a local combinatorial search that is guided by a label switching procedure. Unlike the TSVM, the Laplacian SVM focuses on regularization in reproducing Kernel Hilbert spaces and only needs to solve one small SVM with the labeled data. But Laplacian SVM utilizes heat kernel weights to form edge weights when constructing data adjacency graph. The performance of heat kernel weights highly depends on parameter selection and how to exactly fix parameter in different applications may be troublesome.

Currently, Defense Advanced Research Projects Agency (DARPA) is soliciting innovative research proposals in the area of machine learning for electronic warfare applications and sets up the Behavioral Learning

[*] Corresponding author e-mail: yangjunan@ustc.edu.

for Adaptive Electronic Warfare (BLADE) program [6] in 2010. At the same time, more and more research fruits hold the viewpoint that human behavioral learning can effectively improve the performance of machine learning [7,8,9]. Inspired by these booming trends, we propose a novel approach called Local Behavioral based Laplacian SVM (LB-LapSVM) to overcome the problem of parameter selection in Laplacian SVM.

The rest of this paper is organized as follows. Section 2 briefly describes the semi-supervised learning framework and its extension in Laplacian SVM. Section 3 presents the local behavioral searching method and its utility in LB-LapSVM. Section 4 performs extensive experiments. Finally, we provide some concluding remarks in Section 5.

2 Related work

Inspired by the success of large margin principle, S3VMs are extensions of supervised SVMs to semi-supervised learning by simultaneously learning the optimal hyperplane and the labels for unlabeled instances. It was disclosed that S3VMs realize the low-density assumption by favoring the decision boundary going across low-density regions.

2.1 semi-supervised learning framework

Formally, considering binary classification in semi-supervised learning, we are given a set of l labeled samples $\{\mathbf{x}_i, y_i\}_{i=1}^{l}$, and a set of u unlabeled samples $\{\mathbf{x}_i\}_{i=l+1}^{l+u}$, where $\mathbf{x}_i \in \mathbb{R}^N$ and $y_i \in \{-1,+1\}$. Let us now assume a general-purpose decision function f. The regularized functional to be minimized is defined as:

$$f^* = \arg\min_{f\in\mathscr{H}_K} \frac{1}{l}\sum_{i=1}^{l} V(\mathbf{x}_i, y_i, f) + \gamma_A \|f\|_K^2 + \gamma_I \|f\|_I^2 \quad (1)$$

where V represents a generic cost function of the committed errors on the labeled samples, $\mathscr{H}_K$ is a reproducing kernel Hilbert space (RKHS) induced by the kernel. γ_A controls the complexity of f in the associated Hilbert space $\mathscr{H}_K$, and γ_I controls its complexity in the intrinsic geometry of the marginal data distribution. For example, if the probability distribution is supported on a low-dimensional manifold, $\|f\|_I^2$ penalizes f along that manifold I. Note that this functional constitutes a general regularization framework that takes into account all the available knowledge.

2.2 Laplacian SVM

The previous semi-supervised learning framework allows us to develop many different algorithms just by playing around with the loss function, V, and the regularizes, $\|f\|^2$. In this paper, we focus on the Laplacian SVM formulation, which basically uses a SVM as the learner core and the graph Laplacian for manifold regularization.

A. Cost function of the errors

The Laplacian SVM uses the same hinge loss function as the traditional SVM:

$$V(\mathbf{x}_i, y_i, f) = \max\{0, 1 - y_i f(\mathbf{x}_i)\} \quad (2)$$

where f represents the decision function implemented by the selected classifier.

B. Decision function

We use as the decision function $f(\mathrm{x}_*) = \langle \mathrm{w}, \phi(\mathrm{x}_*)\rangle + b$, where $\phi(\cdot)$ is a nonlinear mapping to a higher (possibly infinite) dimensional Hilbert space $\mathscr{H}$, and $\mathbf{w}$ and b define a linear regression in that space. By means of the Representer Theorem [10], weights $\mathbf{w}$ can be expressed in the dual problem as the expansion over labeled and unlabeled samples $\mathbf{w} = \sum_{i=1}^{l+u} \alpha_i \phi(\mathbf{x}_i) = \boldsymbol{\Phi\alpha}$, where $\boldsymbol{\Phi} = [\phi(\mathbf{x}_1), ..., \phi(\mathbf{x}_{l+u})]^T$ and $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_{l+u}]$. Then, the decision function is given by:

$$f(\mathbf{x}_*) = \sum_{i=1}^{l+u} \alpha_i \mathbf{K}(\mathbf{x}_i, \mathbf{x}_*) + b \quad (3)$$

and $\mathbf{K}$ is the kernel matrix formed by kernel functions, $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle$. The key point here is that, without considering the mapping ϕ explicitly, a non-linear classifier can be constructed by selecting the proper kernel. Also, the regularization term can be fully expressed in terms of the kernel matrix and the expansion coefficients:

$$\|f\|_K^2 = \|\mathbf{w}\|^2 = (\boldsymbol{\Phi\alpha})^T(\boldsymbol{\Phi\alpha}) = \boldsymbol{\alpha}^T\boldsymbol{K\alpha} \quad (4)$$

C. Manifold regularization

The geometry of the data is modeled with a graph in which nodes represent both labeled and unlabeled samples connected by weights $\mathbf{W}_{i,j}$, regularizing the graph follows from the smoothness (or manifold) assumption and intuitively is equivalent to penalize the ”rapid changes” of the classification function evaluated between close samples in the graph:

$$\|f\|_I^2 = \frac{1}{(l+u)^2}\sum_{i,j=1}^{l+u} \mathbf{W}_{ij}(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 = \mathbf{f}^T\mathbf{L}\mathbf{f} \quad (5)$$

where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the graph Laplacian, $\mathbf{D}$ is the diagonal degree matrix of $\mathbf{W}$ given by $D_{ii} = \sum_{j=1}^{l+u} \mathbf{W}_{ij}$,

Algorithm 1 A Summary of LapSVM algorithm [4].

```
Input: $l$ labeled examples $\{\mathbf{x}_i, y_i\}_{i=1}^{l}$, $u$ unlabeled examples $\{\mathbf{x}_i\}_{i=l+1}^{l+u}$
Output: Estimated function $f : \mathbb{R}^N \to \mathbb{R}$
1: Construct data adjacency graph with $(l+u)$ nodes using graph kernel. Choose edge weights through heat kernel weights $\mathbf{W}_{ij} = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/4t}$.
2: Choose a kernel function $\mathbf{K}(\mathbf{x}, y)$. Compute the Gram matrix $\mathbf{K}_{ij} = \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j)$.
3: Compute graph Laplacian matrix: $\mathbf{L} = \mathbf{D} - \mathbf{W}$ where $\mathbf{D}$ is a diagonal matrix given by $D_{ii} = \sum_{j=1}^{l+u} \mathbf{W}_{ij}$.
4: Choose $\gamma_A$ and $\gamma_I$.
5: Compute $\alpha^*$ with the SVM QP solver for soft margin loss.
6: Output function $f^*(x) = \sum_{i=1}^{l+u} \alpha_i^* \mathbf{K}(\mathbf{x}_i, x)$.
```

and $\mathbf{f} = [f(\mathbf{x}_1), ..., f(\mathbf{x}_{l+u})]^T = \mathbf{K}\boldsymbol{\alpha}$, where we have deliberately dropped the bias term b.

The LapSVM algorithm is summarized in the Algorithm 1. From Algorithm 1 we can clearly see that when LapSVM computes edge weights, heat kernel weights $\mathbf{W}_{ij} = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/4t}$ are used. The value of t varies in different applications. The performance of LapSVM will be badly hurt if the value of t is falsely set. Whats more, when computing edge weights, heat kernel weights only focus on sample $\mathbf{x}_i$ and $\mathbf{x}_j$. There is plenty additional information to improve affinity measuring. From the above analysis, we proposed a novel method called LB-LapSVM.

3 Local behavioral searching

In many real world situations, human are exposed to a combination of labeled data and far more unlabeled data when they need to make a classification decision. Understanding how human combine information from labeled and unlabeled data to draw inferences about conceptual boundaries can have significant social impact. In the realistic setting where labeled and unlabeled data is available, semi-supervised learning offers very explicit computational hypotheses that can be empirically tested in the laboratory. To help understand description in this section, we start by providing a translation of relevant terms from semi-supervised learning to human behavioral learning.

That is, when stimulus arrives, human use their supervised experiences from teachers and passive experiences from nature to complete concept learning task. During the learning task, human take in to account of some mechanism in their mind. When concept category is obtained, the learning task is done.

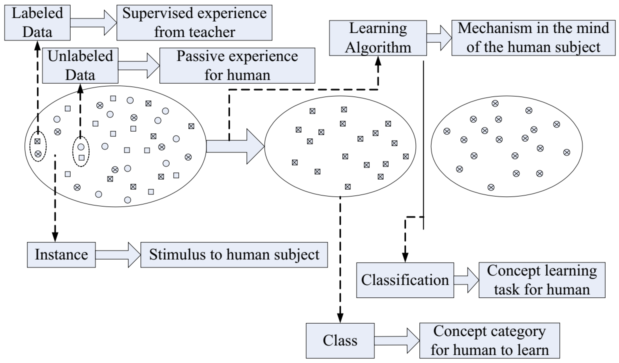

Labeled Data
Supervised experience from teacher
Learning Algorithm
Mechanism in the mind of the human subject
Unlabeled Data
Passive experience for human
Instance
Stimulus to human subject
Classification
Concept learning task for human
Class
Concept category for human to learn

Fig. 1: Translation from semi-supervised learning to human behavioral learning.

Inspired by above analysis, we applied human behavioral learning strategy to LapSVM. In detail, when computing affinity between sample $\mathbf{x}_i$ and $\mathbf{x}_j$, LapSVM only focuses on sample $\mathbf{x}_i$ and $\mathbf{x}_j$. However, its not the case in human behavior. Given a pair of samples, in human empirical cognition, local neighborhood of this pair samples plays an important role in affinity measuring. Based on local distribution of feature space, human behavioral paradigm considers the form of the underlying probability distribution in the neighborhood of a point.

Instead of selecting a single parameter t in heat kernel weights, we propose to calculate local behavioral parameters σ_i and σ_j for data points $\mathbf{x}_i$ and $\mathbf{x}_j$. The distance from $\mathbf{x}_i$ to $\mathbf{x}_j$ as ”seen” by $\mathbf{x}_i$ is $d(\mathbf{x}_i, \mathbf{x}_j)/\sigma_i$ while the converse is $d(\mathbf{x}_j, \mathbf{x}_i)/\sigma_j$. Therefore the square distance d^2 may be generalized as $d(\mathbf{x}_i, \mathbf{x}_j)d(\mathbf{x}_j, \mathbf{x}_i)/\sigma_i\sigma_j = d^2(\mathbf{x}_i, \mathbf{x}_j)/\sigma_i\sigma_j$. The affinity between a pair of points can thus be written as:

$$\mathbf{W}_{i,j} = \exp\left(\frac{-d^2(\mathbf{x}_i, \mathbf{x}_j)}{\sigma_i\sigma_j}\right) \tag{6}$$

Using a specific parameter for each point allows local behavioral searching of the point-to-point distances according to the local statistics of the neighborhoods surrounding points i and j. The selection of local behavioral parameter σ_i can be done by studying the local statistics of the neighborhood of point $\mathbf{x}_i$. In order to catch human thought in neighborhood searching, we choose:

$$\sigma_i {=} d(\mathbf{x}_i, \mathbf{x}_K) \tag{7}$$

where $\mathbf{x}_K$ is the K-th neighbor of point $\mathbf{x}_i$. The selection of K is independent and is a function of the data dimension of the embedding space. Distance metric $d(\mathbf{x}_i, \mathbf{x}_j)$ is set the same to LapSVM, e.g. $\left\|x_i - x_j\right\|$. So based on local behavioral searching, edge weight between $\mathbf{x}_i$ and $\mathbf{x}_j$ can be defined as:

$$\mathbf{W}_{i,j} = \exp\left(\frac{-\left\|x_i - x_j\right\|^2}{\sigma_i\sigma_j}\right) \tag{8}$$

We provide a visualization of the effect of the suggested local behavioral searching strategy in Fig. 2.

Without loss of generality, binary classification is taken into account. In Fig. 2(a), we show input data

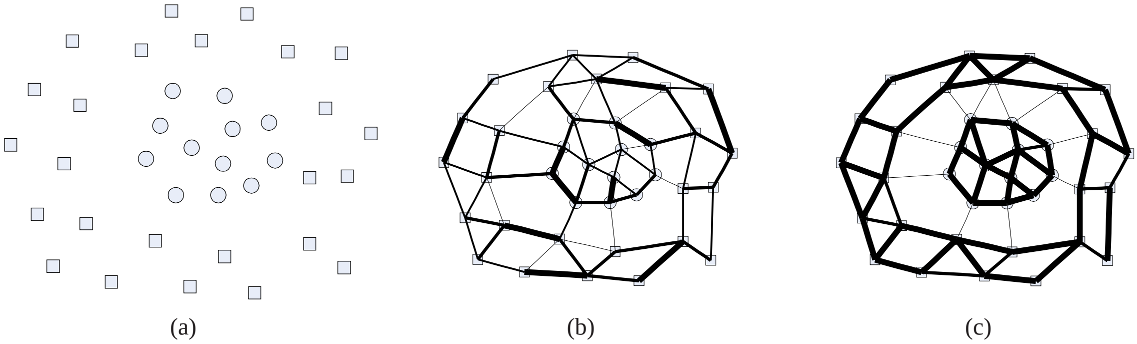


Fig. 2: The effect of local behavioral searching strategy in LapSVM. (a) Input data points. (b) Heat kernel weights based affinity measuring. (c) Local behavioral searching based affinity measuring.

points. Squares and triangles each represent one class. They appear as the shape of a tight cluster resides within a background cluster. Fig. 2(b) shows the affinity measuring results based on heat kernel weights. The affinity between each point and its surrounding neighbors is indicated by the thickness of the line connecting them. Generally, the affinities across tight clusters are larger than the affinities within the sparse background cluster. We can clearly discover that its not the case in Fig. 2(b). Thats to say, the standard approach to estimating affinities fails to capture the data structure. Fig. 2(c) represents the corresponding visualization of affinities after local behavioral searching. The affinities across clusters are now significantly lower than the affinities within any single cluster. Local behavioral searching approach automatically finds the two scales and results in high affinities within clusters and low affinities across clusters. This is the information required for separation.

The local behavioral based LapSVM (LB-LapSVM) algorithm is summarized in the Algorithm 2.

Algorithm 2 A Summary of LB-LapSVM algorithm.

```
Input: $l$ labeled examples $\{\mathbf{x}_i, y_i\}_{i=1}^{l}$, $u$ unlabeled examples $\{\mathbf{x}_i\}_{i=l+1}^{l+u}$
Output: Estimated function $f : \mathbb{R}^N \to \mathbb{R}$
1: Construct data adjacency graph with $(l+u)$ nodes using graph kernel.
2: Compute local behavioral parameters $\sigma_i$ and $\sigma_j$ through formula (7) and build the affinity matrix $\mathbf{W}$ defined by formula (8).
3: Compute graph Laplacian matrix: $\mathbf{L} = \mathbf{D} - \mathbf{W}$ where $\mathbf{D}$ is a diagonal matrix given by $D_{ii} = \sum_{j=1}^{l+u} \mathbf{W}_{ij}$.
4: Choose $\gamma_A$ and $\gamma_I$.
5: Compute $\alpha^*$ with the SVM QP solver for soft margin loss.
6: Output function $f^*(x) = \sum_{i=1}^{l+u} \alpha_i^* \mathbf{K}(\mathbf{x}_i, x)$.
```

4 Experiments

We performed experiments on two moons data set and nine UCI data sets. Based on a survey of related approaches, we chose to compare the proposed LB-LapSVM strategy with SVM (using labeled data only), TSVM, LapSVM. Other approaches lack out-of-sample extension, use different base-classifiers or paradigms, or are implementationally not preferable.

We use classification average error rate as the evaluation measure. The results are reported over 20 runs of each experiment, with different subsets of training and testing data. To measure the inductive performance, we randomly split the data set into two halves and call them the training and test sets. For each data set, 25% data are kept aside as test set to evaluate the performance of learned hypothesis, while the remaining 75% data are training set, which is partitioned into original labeled and unlabeled set.

4.1 Two moons data set

The two moons data set contains 200 points belonging to two non-linearly separable classes with only 1 labeled example for each class. The RBF kernels are chosen for each SVM based algorithms. In TSVM, the parameter N which controls the number of positive points in unlabeled set is set equal to that in labeled set. In LapSVM, parameters γ_A and γ_I are fixed as 0.03125 and 1. The value of local searching range in LB-LapSVM is fixed as 7. The best decision surfaces are shown in Fig. 3.

In Fig. 3, the solid rhombus and circle points represent labeled samples in each class, hollow squares are unlabeled samples and black lines represent the best decision surfaces. Fig. 3 demonstrates how SVM and TSVM fail to find the optimal solution. The LapSVM decision boundary seems to be acceptable, but the boundary is also helpless to those most complex areas. The proposed LB-LapSVM approach can effectively

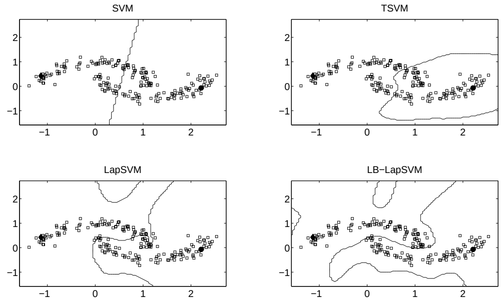


Fig. 3: The best decision surfaces of two moons data set.

discover local intrinsic shape and cause the decision surface to appropriately adjust according to the geometry of the two classes. So the decision surface is intuitively most satisfying.

4.2 UCI data sets

In this section, we evaluate SVM, TSVM, LapSVM with the proposed algorithm on nine UCI data sets shown in Table 1. Since these approaches are applicable for two-class problems, we chose the two-class data sets from UCI data sets. The multiclass data sets in UCI are converted into two-class data sets by choosing the two most populated classes. The data sets are further processed by removing the samples with missing features, if any, binary encoding the categorical features, if any, and reducing the dimensionality using PCA to retain 95 percent of the variance.

Table 1: Experimental UCI data sets.

Data set	Size	Attribute	Class	Pos/Neg
house	232	16	2	46.6%/53.4%
heart	270	9	2	55.6%/44.4%
vehicle	435	26	2	49.9%/50.1%
wdbc	569	14	2	37.3%/62.7%
isolet	600	51	2	50.0%/50.0%
australian	690	14	2	55.5%/44.5%
breastw	699	9	2	65.5%/34.5%
diabetes	768	8	2	34.9%/65.1%
sat	3041	36	2	50.4%/49.6%

The RBF kernels are chosen for each SVM based algorithms. As for these UCI data sets, we randomly select 10, 50 and 100 examples to be used as labeled examples, and use the remaining data as unlabeled data. The experiments are repeated for 20 times and the average error rates are recorded in tables 2 to 4.

Table 2: Average error rate on UCI data sets with 10 labeled examples.

Data set	Average error rate			
	SVM	TSVM	LapSVM	LB-LapSVM
house	0.123	0.112	0.114	**0.106**
heart	0.288	0.265	0.253	**0.242**
vehicle	0.257	**0.212**	0.241	0.227
wdbc	0.209	0.195	0.188	**0.164**
isolet	0.194	0.181	**0.176**	0.179
australian	0.332	0.316	0.311	**0.301**
breastw	0.043	0.041	0.041	**0.038**
diabetes	0.358	0.334	0.317	**0.314**
sat	0.024	0.022	**0.018**	0.019
ave.	0.203	0.186	0.184	**0.177**

Table 3: Average error rate on UCI data sets with 50 labeled examples.

Data set	Average error rate			
	SVM	TSVM	LapSVM	LB-LapSVM
house	0.089	0.067	0.062	**0.055**
heart	0.252	0.246	0.241	**0.228**
vehicle	0.146	**0.122**	0.137	0.128
wdbc	0.135	0.129	0.126	**0.121**
isolet	0.097	0.094	0.088	**0.082**
australian	0.279	0.265	0.233	**0.226**
breastw	0.031	0.029	**0.028**	0.029
diabetes	0.328	0.314	0.305	**0.299**
sat	0.023	0.022	0.018	**0.017**
ave.	0.153	0.143	0.138	**0.132**

Table 4: Average error rate on UCI data sets with 100 labeled examples.

Data set	Average error rate			
	SVM	TSVM	LapSVM	LB-LapSVM
house	0.068	0.047	**0.032**	0.041
heart	0.235	0.231	0.226	**0.217**
vehicle	0.045	0.041	0.038	**0.034**
wdbc	0.078	0.071	0.067	**0.063**
isolet	0.023	0.023	0.020	**0.017**
australian	0.241	0.227	**0.204**	0.209
breastw	0.031	0.028	0.028	**0.026**
diabetes	0.294	0.288	0.284	**0.276**
sat	0.021	0.018	**0.016**	**0.016**
ave.	0.115	0.108	0.102	**0.100**

Tables 2 to 4 summarize the average classification error rates of these algorithms under different numbers of labeled examples. The row *ave.* in each table shows the average results over all the data sets. For each data set row, the biggest improvement percentage has been boldfaced. As can be seen from tables 2 to 4, the proposed LB-LapSVM algorithm achieves highly competitive performance on most data sets under different numbers of labeled examples.

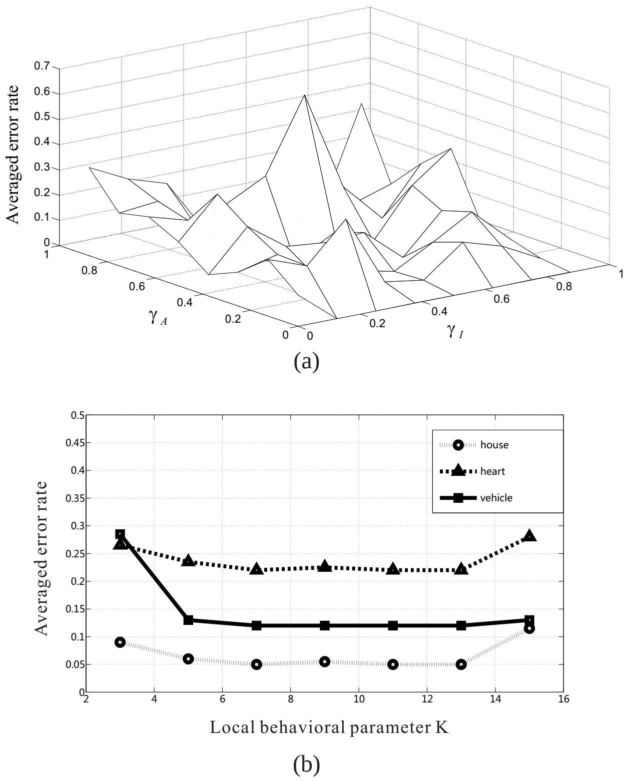
Moreover, tables 2 to 4 also show that if the algorithms are compared through counting the number of winning data sets, i.e., the number of data sets on which one algorithm achieves the best improvement among compared algorithms, LB-LapSVM is always the winner. In detail, under 10 labeled examples, LB-LapSVM has 6 winning sets, while LapSVM and TSVM only have 2 winning sets and 1 wining set respectively; under 50 labeled examples, LB-LapSVM has 7 winning sets, while LapSVM and TSVM only both have 1 winning set respectively; under 100 labeled examples, LB-LapSVM has 7 winning sets, while LapSVM only have 3 winning sets and TSVM with no wining set. So when compared with traditional methods, LB-LapSVM can more effectively and stably exploits unlabeled data to enhance the learning performance.

4.3 Sensitivity to parameter setting

For further studying the performance of the compared semi-supervised learning algorithms, the averaged error rates change of the compared algorithms along with the parameter setting are depicted in Fig. 4. For diversity, two moons data set and three representative UCI data sets with 50 labeled examples are chosen. Regularization parameters γ_A and γ_I are varied in steps in the range $[10^{-6}, 1]$, and local behavioral parameter K is tuned in the range $[3, 5, 7, 9, 11, 13, 15]$. The results are done by cross-validation.

Fig. 4(a) clearly shows that the performance of Laplacian based method strongly depends on setting of regularization parameters γ_A and γ_I. To obtain outstanding results, it is important to search these parameters on the right range. Since γ_A controls the complexity of f in the associated Hilbert space $\mathscr{H}_K$ and γ_I controls its complexity in the intrinsic geometry of the marginal data distribution, we limit the regularization range at $[10^{-6}, 1]$ based on empirical observations.

To perform fully validation with the sensitivity of local behavioral parameter K, three representative UCI data sets are chosen. Results are shown in Fig. 4(b). It is observed that performance will be badly hurt if the value of K is too high or too low. Fixing the value of K at $[7, 11]$ is recommended. The reason can be concluded that local behavioral searching strategy in LapSVM can exactly discover the inner geometric structure between labeled and unlabeled examples. If the value of K is set too low, the local behavioral searching scope can not cover all the affinitive examples and affinity matrix can not fully represent the inner structure between affinitive examples. Contrarily, if the value is fixed beyond normal scope, affinity measuring may suffer interfere from false distribution of adjacency graph and lead to descendent trend of final classification precision.



(a)

(b)

Fig. 4: Illustrative examples of parameter selection for LB-LapSVM. (a) Sensitivity to regularization parameters γ_A and γ_I in two moons data set. (b) Sensitivity to local behavioral parameter K in UCI data sets.

5 Conclusion

We have proposed an algorithm for semi-supervised learning using human behavioral learning framework to solve the confusion of parameter selection in LapSVM. This method brings together the ideas of human learning theory, manifold learning, and kernel-based algorithms in a coherent and natural way to incorporate geometric structure in regularization framework.

A series of experiments has been carried out in order to evaluate performance of the proposed method and compare it with other traditional techniques. From these experiments, it is to be noted that the LB-LapSVM algorithm is effective and stable under different cases.

The main problem encountered is related to the computational cost, since a huge matrix consisting of labeled and unlabeled samples must be inverted. We feel that smart sampling strategies to select the most informative unlabeled samples could yield improve performance, something to be considered in the future.

Acknowledgement

Both authors would like to acknowledge the support of Key Laboratory of Electronic Restriction and National Natural Science Foundation (No. 61179036, No. 60872113).

References

[1] X. J. Zhu, Semi-supervised learning literature survey, Univ. of Wisconsin-Madison, Wisconsin, Tech. Rep. Computer Sciences TR1530, Dec. (2007).

[2] K. Bennett and A. Demiriz, Semi-supervised support vector machines, In Advances in Neural Information Processing Systems, **11**, 368-374 (1999).

[3] T. Joachims, Transductive inference for text classification using support vector machines, Proceedings 16th International Conference on Machine Learning, Bled, Slovenia, 200-209 (1999) .

[4] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: A geometric framework for learning from labeled and unlabeled examples, Journal of Machine Learning Research, **7**, 2399-2434 (2006).

[5] O. Chapelle, A. Zien, Semi-supervised learning by low density separation, Proceedings of the 8th International Conference on Artificial Intelligence and Statistics, 57-64 (2005) .

[6] DARPA, Behavioral Learning for Adaptive Electronic Warfare, DARPA-BAA-10-79, (2010).

[7] T. Mitchell, The discipline of machine learning, Carnegie Mellon University, Tech. Rep. CMU-ML-06-108, (2006).

[8] X.J Zhu, B.R. Gibson, K.S. Jun, Cognitive Models of Test-Item Effects in Human Category Learning, ICML, (2010).

[9] F. Khan, X.J Zhu, B. Mutlu, How Do Humans Teach: On Curriculum Learning and Teaching Dimension, NIPS, (2011).

[10] B. Scholkopf, A. Smola, Learning with Kernels-Support Vector Machines, Regularization, Optimization and Beyond, MIT Press Series, (2002).

Chun Zhang received the M. S. degree from Electronic Engineering Institute, Hefei, China. Currently, he is a Ph.D. candidate in the Department of Computer Engineering at Electronic Engineering Institute, Hefei, China. His research interests have focused on Intelligent Computing, Data Mining and Machine Learning.

Junan Yang received the Ph. D. degree from University of Science and Technology of China. He is currently a professor in the Department of Computer Engineering at Electronic Engineering Institute, Hefei, China. His research interests include Neural Computing, Large-scale machine and Computer Vision.

Jiyang Zhang received the M. S. degree from Electronic Engineering Institute, Hefei, China. He is currently a research engineer at the Jiangnan Institute of Computing Technology, Wuxi, China. His research interests include Computational Intelligence, Signal Processing and Information Security.

Dongsheng Li received the Ph. D. degree from Hefei University of Technology, China. Currently, he is a professor in the Department of Computer Engineering at Electronic Engineering Institute, Hefei, China. His research interests have focused on Intelligent Computing, Multichip Design, Parallel Signal Processing.

Aixia Yong received the Ph. D. degree from Hefei University of Technology, China. She is currently an associate professor in the Department of Computer Engineering at Electronic Engineering Institute, Hefei, China. Her research interests focus on Image Processing, Pattern Recognition and Machine Learning.