Semi-supervised Feature Selection Using Sparse Laplacian Support Vector Machine

Li Zhang[1,2(✉)], Xiaohan Zheng[1], and Zhiqiang Xu[1]

[1] School of Computer Science and Technology, Joint International Research Laboratory of Machine Learning and Neuromorphic Computing, Soochow University, Suzhou 215006, Jiangsu, China
zhangliml@suda.edu.cn

[2] Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou 215006, Jiangsu, China

Abstract. Semi-supervised feature selection is an active topic in machine learning and data mining. Laplacian support vector machine (LapSVM) has been successfully applied to semi-supervised learning. However, LapSVM cannot be directly applied to feature selection. To remedy it, we propose a sparse Laplacian support vector machine (SLapSVM) and apply it to semi-supervised feature selection. On the basis of LapSVM, SLapSVM introduces the ℓ_1-norm regularization, which means the solution of SLapSVM has sparsity. In addition, the training procedure of SLapSVM can be formulated as solving a quadratic programming problem, which indicates that the solution of SLapSVM is unique and global. SLapSVM can perform feature selection and classification at the same time. Experimental results on semi-supervised classification problems show the feasibility and effectiveness of the proposed semi-supervised learning algorithms.

Keywords: Support vector machine · Semi-supervised learning · Feature selection · ℓ_1-norm regularization · Quadratic programming

1 Introduction

Recently, semi-supervised feature selection has attracted substantial attention in machine learning and data mining [23,24]. There are two reasons. One reason is that data collected from real-world applications would have a lot of features. In this case, it is necessary to reduce dimension to achieve better learning performance. As a technique for dimension reduction, feature selection has always been

This work was supported in part by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant No. 19KJA550002, by the Six Talent Peak Project of Jiangsu Province of China under Grant No. XYDXX-054, by the Priority Academic Program Development of Jiangsu Higher Education Institutions, and by the Collaborative Innovation Center of Novel Software Technology and Industrialization.

H. Zhang et al. (Eds.): NCAA 2020, CCIS 1265, pp. 107–118, 2020.
https://doi.org/10.1007/978-981-15-7670-6_10

of concern. The other reason is that labeling examples is expensive and time-consuming while there are large numbers of unlabeled examples available in many practical problems, which results in semi-supervised learning methods. In semi-supervised learning, algorithms construct their models from a few labeled examples together with a large collection of unlabeled data, including graph-based methods [1,4,11,13,30,31], methods based on support vector machine (SVM) [2,3,6,8], and others [5,14,16,19,20,25,26]. This paper focuses on SVM-based methods and discusses the issue of semi-supervised feature selection.

Famous semi-supervised methods based on SVM include transductive support vector machine (TSVM) [8], semi-supervised support vector machine (S3VM) [3], and Laplacian support vector machine (LapSVM) [2]. Bennett et al. proposed S3VM to construct an SVM using both the labeled and unlabeled data [3]. S3VM is iteratively tagging unlabeled data in the training procedure and is usually time consuming. Due to its way to utilize the unlabeled data, S3VM cannot directly classify unseen instances. To implement feature selection using S3VM, Hoai et al. proposed sparse semi-supervised SVM (S4VM) replacing ℓ_2-norm by ℓ_0-norm in S3VM. The objective of S4VM is solved by applying DC (difference of convex) programming [12]. Moreover, Lu et al. cast semi-supervised learning into an ℓ_1-norm linear reconstruction problem and presented an ℓ_1-norm semi-supervised learning method [14]. However, these methods cannot classify new instances directly due to their "closed" nature. LapSVM, an extension of SVM to the semi-supervised field, introduces an additional regularization term on the geometry of both labeled and unlabeled samples by using a graph Laplacian [2]. LapSVM follows a non-iterative optimization procedure and can be taken as a kind of graph-based methods. Gasso et al. proposed an ℓ_1-norm constraint Laplacian SVM (ℓ_1-NC LapSVM) by adding an extra ℓ_1-norm constraint to the optimization problem of LapSVM [9]. The sparseness of the solution to ℓ_1-NC LapSVM is determined by the size of regularization parameter. However, experimental results show that the sparseness of ℓ_1-NC LapSVM is limited for feature selection.

In fact, real data often contains noise, including redundant features, which would have a negative effect on the model performance. In order to eliminate the effect of noise or redundancy on data, it is necessary to generate a sparse decision model to implement feature selection. To implement it, this paper proposes a sparse Laplacian support vector machine (SLapSVM) to perform feature selection. To get a sparse decision model, we adopt the hinge loss and ℓ_1-norm regularization simultaneously. It is known that the hinge loss can lead to a sparse model representation for SVM [15,18]. In addition, the ℓ_1-norm regularization penalty as a substitution of the ℓ_2-norm regularization penalty can also induce a sparse solution [10,21,27,32]. Through the sparse decision model, of SLapSVM, we achieve feature selection. Similar to LapSVM, SLapSVM can be formulated as a quadratic programming problem, which indicates that its solution is unique and global.

The rest of the paper is outlined as follows. Section 2 presents SLapSVM. Section 3 shows experimental results on real-world datasets. Section 4 concludes and discusses further work.

2 SLapSVM

In this section, we propose the model of SLapSVM for semi-supervised learning. For a semi-supervised classification problem, suppose that we have a data set which consists of ℓ labeled and u unlabeled examples. Let $X_\ell = \{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}$ be the labeled set with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{+1, -1\}$, and $X_u = \{\mathbf{x}_i\}_{i=\ell+1}^{\ell+u}$ be the unlabeled set with $\mathbf{x}_i \in \mathbb{R}^d$, where d is the number of features. To integrate these two sets, let $X = \{\mathbf{x}_i\}_{i=1}^{\ell+u}$ be the instance set and $Y = \{y_i\}_{i=1}^{\ell}$ be the label set. Without loss of generalization, the first ℓ examples in the set X correspond to the labeled ones.

The goal of SLapSVM is to find an optimal decision function (model) f from a set of linear hypothesis functions

$$F = \{f(\mathbf{x}) | f(\mathbf{x}) = \boldsymbol{\alpha}^T \mathbf{x} + b, \boldsymbol{\alpha} \in \mathbb{R}^d, b \in \mathbb{R}\} \tag{1}$$

where $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_d]^T$ is the weight vector, and b is the bias.

To obtain the hypothesis function, we replace the ℓ_2-norm regularization in LapSVM by the ℓ_1-norm regularization, and propose LapSVM, which solves the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\alpha}, b, \boldsymbol{\xi}} \quad & \frac{1}{\ell} \sum_{i=1}^{\ell} \xi_i + \gamma_A (\|\boldsymbol{\alpha}\|_1 + \sigma \|b\|_1) + \frac{\gamma_I}{2} \sum_{i=1}^{\ell+u} \sum_{j=1}^{\ell+u} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 W_{ij} \\ s.t. \quad & y_i (\boldsymbol{\alpha}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \ \xi_i \geq 0, \ i = 1, \cdots, \ell \end{aligned} \tag{2}$$

where $\|\cdot\|_1$ is the ℓ_1-norm, $\boldsymbol{\xi} = [\xi_1, \cdots, \xi_\ell] \in \mathbb{R}^\ell$ is the slack vector for labeled samples, W_{ij} is the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, σ is a small positive constant to ensure a unique solution, $\gamma_A \geq 0$ and $\gamma_I \geq 0$ are the regularization parameters. The first term in Eq. (2) is the hinge loss function that is very popular in SVM-like methods and can induce sparsity in theory. The second term $\|\boldsymbol{\alpha}\|_1$ is the ℓ_1-norm regularization term that can also induce sparsity in the ℓ_1-norm SVM [27,32] and sparse signal reconstruction methods [27]. The third term is the Laplacian regularization.

Next, we rewrite the formula Eq. (2) to solve it easily. Since there is no constrain on $\boldsymbol{\alpha}$, the absolute value sign would exist in the objective function Eq. (2) when calculating $\|\boldsymbol{\alpha}\|_1$. In this case, it is not easy to solve Eq. (2). We introduce two vectors $\boldsymbol{\alpha}^+$ and $\boldsymbol{\alpha}^-$ with positive elements, and let

$$\boldsymbol{\alpha} = \boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^- \tag{3}$$

Similarly, we define

$$b = b^+ - b^- \tag{4}$$

where $b^+ > 0$ and $b^- > 0$. In addition, the third term in Eq. (2) can be expressed as:

$$\begin{aligned}\frac{1}{2}\sum_{i,j=1}^{\ell+u}(f(\mathbf{x}_i)-f(\mathbf{x}_j))^2 W_{ij} &= \begin{pmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_{\ell+u}) \end{pmatrix}^T (\mathbf{D}-\mathbf{W}) \begin{pmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_{\ell+u}) \end{pmatrix} \\ &= \begin{pmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_{\ell+u}) \end{pmatrix}^T \mathbf{L} \begin{pmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_{\ell+u}) \end{pmatrix} \\ &= \mathbf{f}^T\mathbf{L}\mathbf{f} \\ &= \boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{L}\mathbf{X}\boldsymbol{\alpha}\end{aligned} \tag{5}$$

where $\mathbf{X} \in \mathbb{R}^{(\ell+u)\times d}$ is the sample matrix in which $\mathbf{x}_i$ is the i-th row, the graph Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$, and $\mathbf{D} \in \mathbb{R}^{(\ell+u)\times(\ell+u)}$ is the diagonal matrix given by $D_{ii} = \sum_j W_{ij}$. Substituting Eqs. (3) and (5) into Eq. (2), we have the following programming:

$$\begin{aligned}\min_{\boldsymbol{\alpha}^\pm, b^\pm, \boldsymbol{\xi}} \quad & \frac{1}{\ell}\sum_{i=1}^{\ell}\xi_i + \gamma_A\left(\sum_{j=1}^{d}\left(\alpha_j^+ + \alpha_j^-\right) + \sigma(b^+ + b^-)\right) + \gamma_I\boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{L}\mathbf{X}\boldsymbol{\alpha} \\ s.t \quad & y_i\left(\mathbf{x}_i^T(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-) + (b^+ - b^-)\right) \geq 1 - \xi_i \\ & b^+, b^- \geq 0, \alpha_j^+, \alpha_j^- \geq 0, j = 1, \cdots, d \\ & \xi_i \geq 0, i = 1, \cdots, \ell\end{aligned} \tag{6}$$

The programming Eq. (6) can be rewritten in matrix form:

$$\begin{aligned}\min_{\mathbf{u}} \quad & \mathbf{c}^T\mathbf{u} + \frac{1}{2}\mathbf{u}^T\mathbf{Q}\mathbf{u} \\ s.t. \quad & \mathbf{A}^T\mathbf{u} \geq \mathbf{1} \\ & \mathbf{u} \geq \mathbf{0}\end{aligned} \tag{7}$$

where $\mathbf{u} = [(\boldsymbol{\alpha}^+)^T, (\boldsymbol{\alpha}^-)^T, b^+, b^-, \boldsymbol{\xi}^T]^T \in \mathbb{R}^{2d+\ell+2}$, $\mathbf{0}$ is the column vector of all zeros, $\mathbf{c} = [\gamma_A\mathbf{1}^T, \gamma_A\mathbf{1}^T, \sigma, \sigma, \mathbf{1}^T/\ell]^T$, $\mathbf{A} = [\mathbf{Y}\mathbf{X}_\ell, -\mathbf{Y}\mathbf{X}_\ell, \mathbf{y}, -\mathbf{y}, \mathbf{I}] \in \mathbb{R}^{\ell\times(2d+\ell+2)}$, $\mathbf{y} = [y_1, y_2, \cdots, y_\ell]^T$, $\mathbf{Y}$ is the diagonal matrix with the diagonal line of $\mathbf{y}$, $\mathbf{1}$ is the column vector of all ones, $\mathbf{I}$ is the $\ell \times \ell$ identity matrix, $\mathbf{X}_\ell$ is the sample matrix of labeled examples, and

$$\mathbf{Q} = \begin{pmatrix} \gamma_I\mathbf{X}^T\mathbf{L}\mathbf{X} & -\gamma_I\mathbf{X}^T\mathbf{L}\mathbf{X} & \mathbf{0} \\ -\gamma_I\mathbf{X}^T\mathbf{L}\mathbf{X} & \gamma_I\mathbf{X}^T\mathbf{L}\mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}.$$

Obviously, Eq. (7) is a constrained quadratic program problem that has $(2d + \ell + 2)$ variables and ℓ inequality constraints. Because the matrix $\mathbf{Q}$ is symmetric and positive semi-definite, this optimization problem could be solved

Algorithm 1: SLapSVM

Input: Instance set $X = \{\mathbf{x}_i\}_{i=1}^{\ell+u}$ and label set $Y = \{y_i\}_{i=1}^{\ell}$, where the first ℓ examples in X have labels corresponding to ones in Y, regularization parameters γ_A and γ_I.

Output: Sparse weight vector $\boldsymbol{\alpha}$ and bias b.

1 **begin**

2 Construct the similarity matrix $\mathbf{W}$:

$$W_{ij} = \begin{cases} \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|_2^2), & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are neighbors} \\ 0, & \text{otherwise} \end{cases}$$

3 Represent matrices $\mathbf{Q}$ and $\mathbf{A}$, and the vector $\mathbf{c}$;

4 Solve the quadratic programming Eq. (7) to obtain the solution $\mathbf{u}$;

5 Get $\boldsymbol{\alpha}^+$, $\boldsymbol{\alpha}^-$, b^+ and b^- from $\mathbf{u}$;

6 Return $\boldsymbol{\alpha} = \boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-$ and $b = b^+ - b^-$.

7 **end**

efficiently through some standard techniques, such as the active set. The algorithm description of SLapSVM is given in Algorithm 1. Step 2 is to construct the similarity matrix, where the parameter γ could be determined by applying the median method used in [28,29].

Once we have $\boldsymbol{\alpha}$ and b, we can obtain the classification hyperplane. For an unseen sample $\mathbf{x}$, SLapSVM predicts its label by

$$\hat{f}(\mathbf{x}) = sign\left(\boldsymbol{\alpha}^T \mathbf{x} + b\right) \tag{8}$$

where $sign(\cdot)$ is the sign function, where $\hat{f}(\mathbf{x})$ is the estimated label for the unseen sample $\mathbf{x}$.

Let $NZ = \{\alpha_i | \alpha_i \neq 0, i = 1, \cdots, d\}$ be the set of non-zero coefficients for Eq. 8, where $|\cdot|$ is the number of elements in a set. Because both the hinge loss and the ℓ_1-norm can induce sparsity, we could get a sparse vector $\boldsymbol{\alpha}$ that corresponds to weights of features. Thus, the inequality $|NZ| < d$ holds true, and we can perform the operation of feature selection. The set NZ can actually reflect the selected feature subset and show the sparsity of the decision model. The smaller $|NZ|$ is, the more sparsity the decision model has.

3 Experimental Results

In this section, we validate the effectiveness of the proposed method in feature selection on synthetic and UCI [7] datasets. To demonstrate the capabilities of our algorithm, this paper compares SLapSVM with the state-of-art algorithms for feature selection, including S3VM-PiE [12], S3VM-PoDC [12], S3VM-SCAD [12], S3VM-Log [12], S3VM-ℓ_1 [12], and Lap-PPSVM [22]. All numerical experiments are performed on a personal computer with an Inter Core I5 processor with 4 GB RAM. This computer runs Windows 7, with Matlab R2013a.

3.1 Data Description and Experimental Setting

Datasets used here include a toy and seven UCI ones, which are described as follows:

- Gaussian data
  In the Gaussian dataset, two-class synthetic samples are drawn from two Gaussian distributions: $N((0,0)^T, \mathbf{I})$ and $N((3,0)^T, \mathbf{I})$, where $\mathbf{I} \in \mathbb{R}^{2\times 2}$ is the identify matrix. There are 600 samples total and 300 samples for each class. For each class, 80% of data are selected as the training samples and the rest as the test ones.
- UCI data
  Seven UCI datasets are summarized in Table 1. These datasets represent a wide range of fields (including pathology, vehicle engineering, biological information, finance and so on), sizes (from 267 to 1473) and features (from 9 to 34). All datasets are normalized such that the features scale in the interval $[-1, 1]$ before training and test. Similar to [22], in our experiments, each UCI dataset is divided into two subsets randomly: 70% for training and 30% for test.

When we compare the different methods, some performance indexes would be considered, such as accuracy, F1-measure, and sparsity. These three performance indexes are described as follows.

- Accuracy is defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

  where TP means the number of true positive samples, TN means the number of true negative samples, FP means the number of false positive samples and FN means the number of false negative samples.
- F1-measure can be defined as:

$$F1 - measure = \frac{2P \times R}{P + R} \tag{10}$$

  where $P = TP/(TP + FP)$ is precision and $R = TP/(TP + FN)$ is recall.
- Sparsity is measured by $|NZ|$.

All regularization parameters in compared methods are selected from the set $\{10^{-6}, \cdots, 10^{2}\}$ using two-fold cross-validation [17]. Once the parameters are selected, they would be returned to the training subset to learn the final decision function. Each experiment is repeated 10 times and the average results on test subsets are reported.

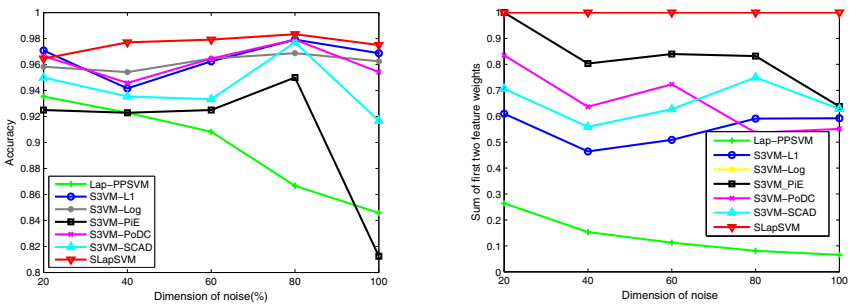
Table 1. Description of seven UCI datasets

	Australian	CMC	German	Ionosphere	Hearts	Spect	WDBC
#Sample	700	1473	1000	351	270	267	569
#Attribute	14	9	24	34	13	22	14
#Class	2	2	2	2	3	2	2

3.2 Gaussian Data

Consider the random Gaussian dataset. In the training subset, we randomly take 10% as the labeled set and the rest as the unlabeled set. In order to verify the ability to select features, we append m-dimensional noise to the training subset, where m takes a value in the set $\{20, 40, 60, 80, 100\}$. The noise in each dimension is the white Gaussian noise and has a signal-noise ratio (SNR) of 3 dB. Note that the original features are the first two ones in the $(m + 2)$-dimensional dataset. Consider the variation of m, we choose the accuracy and the sum of the first two feature weights as the metrics to compare SLapSVM with other methods.

The average experimental results are given in Fig. 1. Basically, SLapSVM always achieves the best average accuracy when $m > 20$, as shown in Fig. 1(a). For visualization, we normalize the weight vector so that the sum of all weights is equal to one. From Fig. 1(b), we can see the good performance of SLapSVM in eliminating noise or the good ability to select useful features. Only can SLapSVM pick up the first two useful features. In other words, SLapSVM can accurately select those features that are helpful for classification.



(a) Classification accuracy (b) Sum of first two feature weights

Fig. 1. Performance vs. dimension of noise on Gaussian dataset, (a) classification accuracy and (b) sum of first two feature weights.

Further, we list the best performance and the corresponding weights of all methods in Table 2, where the best accuracy among these compared methods is in bold type, and "First weight" and "Second weight" mean that weights of the first and the second features, respectively. From Table 2, we can see that

SLapSVM has the best accuracy. Also, weights of the first and the second feature are evenly distributed. Note that the weight vector has been normalized, or the sum of all weights is equal to one.

Table 2. Comparison of seven methods on the Gaussian dataset

Methods	Accuracy(%)	First weight	Second weight
S3VM-ℓ_1	97.92 ± 2.26	0.3433	0.2519
S3VM-Log	96.71 ± 1.78	0.3643	0.2892
S3VM-PiE	95.00 ± 2.97	0.8175	0.0279
S3VM-PoDC	97.92 ± 3.77	0.2648	0.2721
S3VM-SCAD	97.71 ± 2.21	0.3342	0.4257
Lap-PPSVM	93.54 ± 1.67	0.1337	0.1461
SLapSVM	**98.33** ± 2.03	0.5298	0.4702

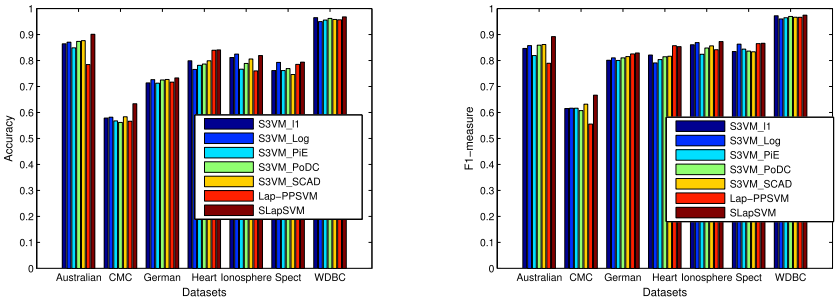
3.3 UCI Datasets

Seven UCI datasets are considered here. In the training subset, we randomly take 40% as the labeled set and the rest as unlabeled set, which follows the setting in [12]. We compare the effectiveness of seven algorithms and report the average results in Fig. 2, where Figs. 2(a), 2(b) and 2(c) show the average accuracy, F1-measure and the number of non-zero coefficients, respectively. Here, the number of non-zero coefficients reflects the ability of feature selection.

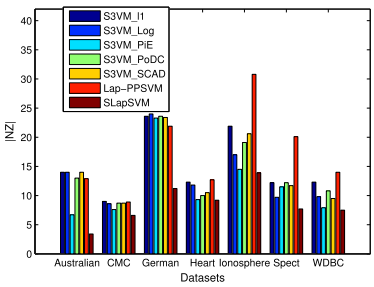
On the index of accuracy Fig. 2(a), SLapSVM performs the best among all seven methods on six out of seven datasets. On the Ionoshpere dataset, SLapSVM is slightly inferior to S3VM-Log. On both Australian and CMC datasets, SLapSVM has a great improvement in classification performance. On the other four datasets, SLapSVM is slightly superior to the compared methods.

On the index of F1-measure Fig. 2(b), SLapSVM also performs the best among all seven methods on six out of seven datasets. On the Heart dataset, SLapSVM is slightly inferior to Lap-PPSVM. On both Australian and CMC datasets, ℓ_1-norm has also a great improvement. On the other four datasets, SLapSVM is slightly superior to the compared methods.

From Figs. 2(a) and 2(b), we can conclude that SLapSVM performs very well compared to other six methods on the performance indexes of accuracy and F1-measure. Moreover, we focus on Fig. 2(c) that shows the ability to select features. We can see that SLapSVM has a significantly higher feature sparsity than other methods while maintaining the high classification performance. In other words, SLapSVM can achieve a better performance using less features.



(a) Accuracy (b) F1-measure



(c) $|NZ|$

Fig. 2. Different performance comparison on seven datasets, (a) accuracy, (b) F1-measure, and (c) $|NZ|$.

3.4 Parameter Sensitivity Analysis

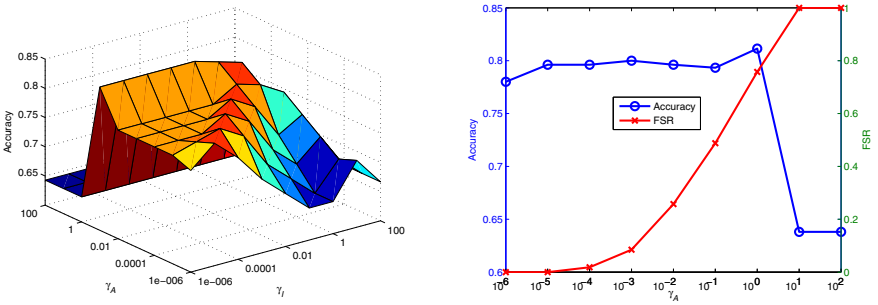
As we can see above, SLapSVM has two parameters γ_A and γ_I. We are interested in the classification performance of our algorithm when the parameters γ_A and γ_I are changed and the sparsity of our algorithm when γ_A varies. In order to measure the sparsity of SLapSVM, we define the feature sparsity ratio (FSR) as follows:

$$FSR = 1 - \frac{|NZ|}{d}$$

where d is the number of features and $0 \leq FSR \leq 1$. $FSR = 0$ means that all features are selected and there has no sparsity, and $FSR = 1$ means that none of features are selected.

For this purpose, we choose the Ionosphere dataset. To observe the effect of regularization parameters on the algorithm performance, we change both γ_A and γ_I from 10^{-6} to 10^2. The resulted curve of accuracy vs. γ_A and γ_I obtained by SLapSVM on the Ionosphere dataset is shown in Fig. 3(a). From this figure, we can see that when γ_A is fixed, SLapSVM can achieve a better accuracy if γ_I is small. For a fixed γ_I, the performance of SLapSVM varies largely with changing γ_A. Thus, an appropriate γ_A would bring a good result.

Further, we analyze the effect of γ_A on the performance of SLapSVM. The curves of both accuracy and FSR vs. γ_A are shown in the left axes and the right axes of Fig. 3(b), respectively. We can observe that as γ_A increases, FSR of SLapSVM is getting greater. The variation of accuracy is slightly complexity. The accuracy corresponding to $0 < FSR < 1$ is greater than the one with $FSR = 0$ or $FSR = 1$. When $FSR = 1$, an arbitrary test sample would be assigned to a positive label. Note that γ_A controls the sparsity of the weight vector, and γ_I the Laplacian regularization term. Thus, the sparsity regularization has a greater effect on the performance than the Laplacian regularization does.



(a) Accuracy (b) Accuracy and FSR

Fig. 3. Performance vs. regularization parameters on Ionoshpere, (a) accuracy, and (b) accuracy and FSR.

4 Conclusions

In this paper, we propose a novel sparse LapSVM for semi-supervised learning by replacing the ℓ_2-norm regularization with the ℓ_1-norm regularization, called SLapSVM. Extensive experiments are conducted to validate the feasibility and effectiveness of SLapSVM on feature selection. Among compared semi-supervised methods based on SVM, SLapSVM has the best ability of feature selection, which can be supported by experimental results on the Gaussian dataset. Furthermore, experimental results on seven UCI datasets also indicate the superiority of the SLapSVM in feature selection and classification.

References

1. Belkin, M., Niyogi, P.: Semi-supervised learning on riemannian manifolds. Mach. Learn. **56**(1–3), 209–239 (2004)
2. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. J. Mach. Learn. Res. **7**(1), 2399–2434 (2006)

3. Bennett, K.P., Demiriz, A.: Semi-supervised support vector machines. In: Proceedings of International Conference on Neural Information Processing Systems, pp. 368–374 (1999)
4. Blum, A., Chawla, S.: Learning from labeled and unlabeled data using graph mincuts. In: Proceedings of Eighteenth International Conference on Machine Learning, pp. 19–26 (2001)
5. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the 1998 Conference on Computational Learning Theory, pp. 92–100 (1998)
6. Cheng, S.J., Huang, Q.C., Liu, J.F., Tang, X.L.: A novel inductive semi-supervised SVM with graph-based self-training. In: Yang, J., Fang, F., Sun, C. (eds.) IScIDE 2012. LNCS, vol. 7751, pp. 82–89. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36669-7_11
7. Dheeru, D., Karra Taniskidou, E.: UCI machine learning repository (2017). http://archive.ics.uci.edu/ml
8. Gammerman, A., Vovk, V., Vapnik, V.: Learning by transduction. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, pp. 148–155. Morgan Kaufmann, San Francisco, CA (2013)
9. Gasso, G., Zapien, K., Canu, S.: Sparsity regularization path for semi-supervised SVM. In: Proceedings of International Conference on Machine Learning and Applications, pp. 25–30 (2007)
10. Jiang, J., Ma, J., Chen, C., Jiang, X., Wang, Z.: Noise robust face image super-resolution through smooth sparse representation. IEEE Trans. Cybern. **47**(11), 3991–4002 (2017)
11. Kothari, R., Jain, V.: Learning from labeled and unlabeled data. In: Proceedings of International Joint Conference on Neural Networks, pp. 2803–2808 (2002)
12. Le, H.M., Thi, H.A.L., Nguyen, M.C.: Sparse semi-supervised support vector machines by DC programming and DCA. Neurocomputing **153**, 62–76 (2015)
13. Liu, W., He, J., Chang, S.F.: Large graph construction for scalable semi-supervised learning. In: Proceedings of International Conference on Machine Learning, pp. 679–686 (2010)
14. Lu, Z., Peng, Y.: Robust image analysis by l1-norm semi-supervised learning. IEEE Trans. Image Process. **24**(1), 176–188 (2015)
15. Poggio, T., Girosi, F.: A sparse representation for function approximation. Neural Comput. **10**(6), 1445–1454 (1998)
16. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)
17. Refaeilzadeh, P., Tang, L., Liu, H.: Cross-Validation. In: Lui, L., Özsu, M.T. (eds.) Encyclopedia of Database Systems, pp. 532–538. Springer, New York (2016)
18. Schölkopf, B.: Sparseness of support vector machines. Mach. Learn. **4**(6), 1071–1105 (2008)
19. Shahshahani, B.M., Landgrebe, D.A.: The effect of unlabeled samples in reducing the small sample size problem and mitigating the hughes phenomenon. IEEE Trans. Geosci. Remoto Sens. **32**(5), 1087–1095 (1994)
20. Shental, N., Bar-Hillel, A., Hertz, T., Weinshall, D.: Gaussian mixture models with equivalence constraints. In: Constrained Clustering: Advances in Algorithms, Theory, and Applications, pp. 33–58. Chapman & Hall, London (2009)
21. Sun, Y., et al.: Discriminative local sparse representation by robust adaptive dictionary pair learning. IEEE Trans. Neural Networks Learn. Syst. 1–15 (2020)

22. Tan, J., Zhen, L., Deng, N., Zhang, Z.: Laplacian p-norm proximal support vector machine for semi-supervised classification. Neurocomputing **144**(1), 151–158 (2014)
23. Tang, B., Zhang, L.: Semi-supervised feature selection based on logistic I-RELIEF for multi-classification. In: Geng, X., Kang, B.-H. (eds.) PRICAI 2018. LNCS (LNAI), vol. 11012, pp. 719–731. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-97304-3_55
24. Tang, B., Zhang, L.: Multi-class semi-supervised logistic I-RELIEF feature selection based on nearest neighbor. In: Yang, Q., Zhou, Z.-H., Gong, Z., Zhang, M.-L., Huang, S.-J. (eds.) PAKDD 2019. LNCS (LNAI), vol. 11440, pp. 281–292. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-16145-3_22
25. Yakowitz, S.J.: An introduction to Bayesian networks. Technometrics **39**(3), 336–337 (1997)
26. Yedidia, J.H.S., Freeman, W.T., Weiss, Y.: Generalized belief propagation. Adv. Neural Inf. Process. Syst. **13**(10), 689–695 (2000)
27. Zhang, L., Zhou, W.: On the sparseness of 1-norm support vector machines. Neural Networks **23**(3), 373–385 (2010)
28. Zhang, L., Zhou, W., Chang, P., Liu, J., Yan, Z., Wang, T., Li, F.: Kernel sparse representation-based classifier. IEEE Trans. Signal Process. **60**, 1684–1695 (2012)
29. Zhang, L., Zhou, W., Li, F.: Kernel sparse representation-based classifier ensemble for face recognition. Multimedia Tools Appl. **74**(1), 123–137 (2015)
30. Zhou, D., Schölkopf, B.: Learning from labeled and unlabeled data using random walks. In: Rasmussen, C.E., Bülthoff, H.H., Schölkopf, B., Giese, M.A. (eds.) DAGM 2004. LNCS, vol. 3175, pp. 237–244. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28649-3_29
31. Zhou, D., Schölkopf, B.: A regularization framework for learning from graph data. In: ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields, pp. 132–137 (2004)
32. Zhu, J., Rosset, S., Hastie, T., Tibshirani, R.: 1-norm support vector machines. In: Proceedings of the 16th International Conference on Neural Information Processing Systems, vol. 16, no. 1, pp. 49–56 (2003)