Journal of Information & Computational Science 8: 6 (2011) 885–892
Available at http://www.joics.com

Semi-supervised Least-squares Support Vector Regression Machines ⋆

Shuo Xu [a], Xin An [b], Xiaodong Qiao [a], Lijun Zhu [a], Lin Li [c,*]

[a] *Information Technology Supporting Center, Institute of Scientific and Technical Information of China Beijing 100038, China*

[b] *School of International Trade and Economics, University of International Business and Economics Beijing 100029, China*

[c] *College of Information and Electrical Engineering, China Agricultural University Beijing 100083, China*

Abstract

In many real-world applications, unlabeled examples are inexpensive and easy to obtain. Semi-supervised approaches try to utilize such examples to boost the predictive performance. But previous research mainly focuses on classification problem, and semi-supervised regression remains largely under-studied. In this work, a novel semi-supervised regression method, semi-supervised LS-SVR (S^2LS-SVR), is proposed on the basis of LS-SVR. Similar to the LS-SVR, one only solves a convex linear system in the training phrase too, thus largely speeding up training. Experimental results on corn data set indicate that our approach is feasible and efficient.

Keywords: Semi-supervised Learning; Regression Problem; Least-squares Support Vector Regression Machine (LS-SVR); Semi-supervised LS-SVR (S^2LS-SVR)

1 Introduction

Traditionally, hypotheses are learned from a large number of training examples, in each of which a *label* is attached. For classification problem, the label indicates the *category* into which the corresponding example falls; for regression problem, the label is a real-value. Most machine learning methods rely on the availability of large labeled examples, since the larger the number

⋆This work was funded partially by the "Key Technologies Research on Large Scale Semantic Computation for Foreign Scientific & Technical Knowledge Organization System", which is sponsored by Key Technologies R&D Program of Chinese 12th Five-year Plan (2011-2013) under grant number 2011BAH10B04, Foundation for the Author of Excellent Doctoral Dissertation of University of International Business under grant under 73600002, and "Special Funds for Agro-Scientific Research in Public Interest in P. R. China" under grant 200903021.

*Corresponding author.

Email addresses: xush@istic.ac.cn (Shuo Xu), anxin927@gmail.com (Xin An), qiaox@istic.ac.cn (Xiaodong Qiao), zhulj@istic.ac.cn (Lijun Zhu), lilincau@gmail.com (Lin Li).

June 2011

of training examples, the better the performance of the resulting machines. However, human annotation is time-consuming, making labeled data costly to obtain in practice.

To overcome this problem, a co-training algorithm is proposed by Blum and Mitchell [1] in 1988. Since then, many *semi-supervised* learning approaches [2, 3, 4] are raised, such as semi-supervised support vector classifier (S^3VC) [5], etc. These methods leverage large amounts of relatively inexpensive unlabeled data along with small amounts of labeled data. The empirical results of these papers indicate that indeed unlabeled data can be used to significantly improve the predictive performance. But previous research mainly focuses on classification problem, and semi-supervised regression remains largely under-studied. To the best of our knowledge, only semi-supervised ridge regression [6, 7] and co-training kNN [8, 9] has been put forward in literatures. What's more, it is difficult to generalize directly semi-supervised classification models to counterpart regression ones [6].

By changing the inequality constraints in the support vector regression machine (SVR) [10, 11] by the equality ones, the least-squares SVR (LS-SVR) [12] replaces convex quadratic programming problem with convex linear system solving problem, thus largely speeding up training. It has been shown through a meticulous empirical study that the generalization performance of the LS-SVR is comparable to that of the SVR [13]. Therefore, the LS-SVR has been attracting extensive attentions during the past few years, such as [14] and references therein. On the basis of the LS-SVR, a novel semi-supervised regression approach, semi-supervised LS-SVR (S^2LS-SVR), is proposed in Section 3. Similar to the LS-SVR, one only solves a convex linear system in the training phrase, too. In Section 4 and Section 5, an experimental evaluation is conducted, and Section 6 concludes this work.

2 Least-squares Support Vector Regression Machine (LS-SVR)

Given a training set $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_m, y_m)\}$ with $(\mathbf{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{R}(i = 1, 2, \cdots, m)$. Let $\mathbf{y} = (y_1, y_2, \cdots, y_m)^{\mathrm{T}}$. Then the primal problem of the LS-SVR can be formally defined as [12]

$$\min_{\mathbf{w}\in\mathbb{R}^{n_h}, b\in\mathbb{R}, \xi\in\mathbb{R}^m} J(\mathbf{w}, \xi) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \gamma\frac{1}{2}\xi^{\mathrm{T}}\xi \tag{1}$$

$$\text{s.t. } y_i = \mathbf{w}^{\mathrm{T}}\varphi(\mathbf{x}_i) + b + \xi_i, i = 1, 2, \cdots, m \tag{2}$$

where $\varphi : \mathbb{R}^d \to \mathbb{R}^{n_h}$ is a mapping to some higher (maybe infinite) dimensional feature space $\mathcal{H}$ with n_h dimensions, each component of $\xi = (\xi_1, \xi_2, \cdots, \xi_m)^{\mathrm{T}}$ is a slack variable, and γ is a positive real regularized parameter.

Through the Karush-Kuhn-Tucker (KKT) conditions of the Lagrangian, the solution of the problem (1)-(2) is the same as that of the following linear system:

$$\begin{bmatrix} 0 & \mathbf{e}^{\mathrm{T}} \\ \mathbf{e} & \mathbf{K} + \gamma^{-1}\mathbf{I} \end{bmatrix} \begin{bmatrix} b & \alpha \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{y} \end{bmatrix} \tag{3}$$

where $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_m)^{\mathrm{T}}$ is a vector consisting of Lagrange multipliers, $\mathbf{e} = (1, 1, \cdots, 1)^{\mathrm{T}}$, $\mathbf{I}$ is an identity matrix and $K_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)^{\mathrm{T}}\varphi(\mathbf{x}_j)(i, j = 1, 2, \cdots, m)$ with $K(\cdot, \cdot)$ is a kernel function meeting the Mercer's theorem [10, 11].

Let the solution of linear system (3) be $\alpha^* = (\alpha_1^*, \alpha_2^*, \cdots, \alpha_m^*)^{\mathrm{T}}$ and b^*. The corresponding decision function is

$$f(\mathbf{x}) = \mathbf{w}^{*\mathrm{T}}\varphi(\mathbf{x}) + b^* = \sum_{i=1}^{m} \alpha_i^* \varphi(\mathbf{x}_i)^{\mathrm{T}}\varphi(\mathbf{x}) + b^* = \sum_{i=1}^{m} \alpha_i^* K(\mathbf{x}_i, \mathbf{x}) + b^*. \tag{4}$$

3 Semi-supervised LS-SVR (S^{2}LS-SVR)

Given a labeled training set $\mathcal{L} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_m, y_m)\}$ and a unlabeled training set $\mathcal{U} = \{\mathbf{x}_{m+1}, \mathbf{x}_{m+2}, \cdots, \mathbf{x}_{m+u}\}$. Usually, these data are at hand and $u \gg m$. For convenience, let $\mathbf{y} = (y_1, y_2, \cdots, y_m)^{\mathrm{T}}$.

3.1 Primal and Dual Problems

Suppose one can estimate the resulting label $\hat{y}_i$ of $\mathbf{x}_{m+i} \in \mathcal{U}(i = 1, 2, \cdots, u)$ in some way. Defining $\hat{\mathbf{y}} = (\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_u)^{\mathrm{T}}$. Similar to semi-supervised ridge regression [6, 7], the primal problem of the S^{2}LS-SVR can be formally defined

$$\min_{\mathbf{w}\in\mathbb{R}^{n_h}, b\in\mathbb{R}, \xi\in\mathbb{R}^m, \zeta\in\mathbb{R}^u} J(\mathbf{w}, \xi, \zeta) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \gamma\frac{1}{2}\xi^{\mathrm{T}}\xi + \lambda\frac{1}{2}\zeta^{\mathrm{T}}\zeta \tag{5}$$

$$\text{s.t. } y_i = \mathbf{w}^{\mathrm{T}}\varphi(\mathbf{x}_i) + b + \xi_i, i = 1, 2, \cdots, m \tag{6}$$

$$\hat{y}_i = \mathbf{w}^{\mathrm{T}}\varphi(\mathbf{x}_{m+i}) + b + \zeta_i, i = 1, 2, \cdots, u \tag{7}$$

where $\xi = (\xi_1, \xi_2, \cdots, \xi_m)^{\mathrm{T}}$ and $\zeta = (\zeta_1, \zeta_2, \cdots, \zeta_u)^{\mathrm{T}}$ are resulting vectors consisting of slack variables, which correspond respectively to the training set $\mathcal{L}$ and $\mathcal{U}$, and γ and λ are positive real regularized parameters as in the LS-SVR case.

The Lagrangian for the problem (5)-(7) is

$$L(\mathbf{w}, b, \xi, \zeta, \alpha, \beta) = J(\mathbf{w}, \xi, \zeta) - \sum_{i=1}^{m} \alpha_i\{\mathbf{w}^{\mathrm{T}}\varphi(\mathbf{x}_i) + b + \xi_i - y_i\} - \sum_{i=1}^{u} \beta_i\{\mathbf{w}^{\mathrm{T}}\varphi(\mathbf{x}_{m+i}) + b + \zeta_i - \hat{y}_i\} \tag{8}$$

where $\alpha_i (i = 1, 2, \cdots, m)$ and $\beta_i (i = 1, 2, \cdots, u)$ values are the Lagrange multipliers. Let $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_m)^{\mathrm{T}}$ and $\beta = (\beta_1, \beta_2, \cdots, \beta_u)^{\mathrm{T}}$. The KKT conditions for optimality yield

$$\begin{cases} \dfrac{\partial L}{\partial \mathbf{w}} = 0 & \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{m} \alpha_i \varphi(\mathbf{x}_i) + \sum_{i=1}^{u} \beta_i \varphi(\mathbf{x}_{m+i}) & \\ \dfrac{\partial L}{\partial b} = 0 & \Rightarrow \quad \sum_{i=1}^{m} \alpha_i + \sum_{i=1}^{u} \beta_i = 0 & \\ \dfrac{\partial L}{\partial \xi_i} = 0 & \Rightarrow \quad \alpha_i = \gamma\xi_i, & i = 1, 2, \cdots, m \\ \dfrac{\partial L}{\partial \zeta_i} = 0 & \Rightarrow \quad \beta_i = \lambda\zeta_i, & i = 1, 2, \cdots, u \\ \dfrac{\partial L}{\partial \alpha_i} = 0 & \Rightarrow \quad \mathbf{w}^{\mathrm{T}}\varphi(\mathbf{x}_i) + b + \xi_i - y_i = 0, & i = 1, 2, \cdots, m \\ \dfrac{\partial L}{\partial \beta_i} = 0 & \Rightarrow \quad \mathbf{w}^{\mathrm{T}}\varphi(\mathbf{x}_{m+i}) + b + \zeta_i - \hat{y}_i = 0, & i = 1, 2, \cdots, u \end{cases} \tag{9}$$

Similar to the LS-SVR, eliminating $\mathbf{w}$, ξ and ζ, one can obtain the following linear system (dual problem):

$$\begin{bmatrix} 0 & \mathbf{e}_m^{\mathrm{T}} & \mathbf{e}_u^{\mathrm{T}} \\ \mathbf{e}_m & \mathbf{K}_{1,1}+\gamma^{-1}\mathbf{I}_m & \mathbf{K}_{1,2} \\ \mathbf{e}_u & \mathbf{K}_{2,1} & \mathbf{K}_{2,2}+\lambda^{-1}\mathbf{I}_u \end{bmatrix} \begin{bmatrix} b \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{y} \\ \hat{\mathbf{y}} \end{bmatrix} \tag{10}$$

where $\mathbf{e}_m$ and $\mathbf{e}_u$ are all-one column vectors with m and u elements, respectively, and $\mathbf{I}_m$ and $\mathbf{I}_u$ are m- and u-order identity matrix, respectively. Let $\mathbf{Z}_1 = (\varphi(\mathbf{x}_1), \varphi(\mathbf{x}_2), \cdots, \varphi(\mathbf{x}_m))$, $\mathbf{Z}_2 = (\varphi(\mathbf{x}_{m+1}), \varphi(\mathbf{x}_{m+2}), \cdots, \varphi(\mathbf{x}_{m+u}))$. Then, $\mathbf{K}_{1,1} = \mathbf{Z}_1^{\mathrm{T}}\mathbf{Z}_1$, $\mathbf{K}_{1,2} = \mathbf{K}_{2,1}^{\mathrm{T}} = \mathbf{Z}_1^{\mathrm{T}}\mathbf{Z}_2$, $\mathbf{K}_{2,2} = \mathbf{Z}_2^{\mathrm{T}}\mathbf{Z}_2$.

Let the solution of linear system (10) be $\alpha^* = (\alpha_1^*, \alpha_2^*, \cdots, \alpha_m^*)^{\mathrm{T}}$, $\beta^* = (\beta_1^*, \beta_2^*, \cdots, \beta_u^*)^{\mathrm{T}}$ and b^*. Then the corresponding decision function is

$$f(\mathbf{x}) = \mathbf{w}^{*\mathrm{T}}\varphi(\mathbf{x}) + b^* = \sum_{i=1}^{m} \alpha_i^* K(\mathbf{x}_i, \mathbf{x}) + \sum_{i=1}^{u} \beta_i^* K(\mathbf{x}_{m+i}, \mathbf{x}) + b^*. \tag{11}$$

3.2 To Estimate the Labels of Each Example in the Training Set $\mathcal{U}$

As mentioned in subsection 3.1, the label of each example in the training set $\mathcal{U}$ is needed in order to solve the linear system (10). Whether semi-supervised classification or regression, it is necessary to estimate the resulting labels in the set $\mathcal{U}$ in some way [3]-[9]. This paper utilizes the neighborhood concept in the feature space $\mathcal{H}$. Specifically, for $\forall \mathbf{x} \in \mathcal{U}$, let $\mathcal{N}_k(\mathbf{x}) = \{\varphi(\mathbf{x}_{i_1}), \varphi(\mathbf{x}_{i_2}), \cdots, \varphi(\mathbf{x}_{i_k}) | \mathbf{x}_{i_t} \in \mathcal{L}(t = 1, 2, \cdots, k)\}$ be the set consisting of k nearest neighborhoods in the feature space $\mathcal{H}$. One can readily estimate the label of $\mathbf{x}$ as the weighted average of the neighborhood labels in $\mathcal{N}_k(\mathbf{x})$, i.e. $\hat{y} = \frac{(\sum_{t=1}^{k} d_t^{-1} y_{i_t})}{\sum_t d_t^{-1}}$, where d_t is the distance between $\varphi(\mathbf{x})$ and $\varphi(\mathbf{x}_{i_t})$ in the feature space $\mathcal{H}$. In fact, the distance d_t can be calculated in the input space by the kernel trick [15] as follows

$$\begin{aligned} \|\varphi(\mathbf{x}) - \varphi(\mathbf{x}_{i_t})\| &= \sqrt{(\varphi(\mathbf{x}) - \varphi(\mathbf{x}_{i_t}))^{\mathrm{T}}(\varphi(\mathbf{x}) - \varphi(\mathbf{x}_{i_t}))} \\ &= \sqrt{K(\mathbf{x}, \mathbf{x}) - 2K(\mathbf{x}_{i_t}, \mathbf{x}) + K(\mathbf{x}_{i_t}, \mathbf{x}_{i_t})}. \end{aligned} \tag{12}$$

4 Experimental Data

Corn data set (http://www.eigenvector.com/data/Corn/index.html) consists of 80 examples of corn measured on 3 different near-infra-red spectrometers, m5, mp5 and mp6. In this study, spectra from instrument m5 are used, where the wavelength range is 1100-2498nm at 2nm intervals. The moisture, oil, protein and starch values represent four output/dependent variables. As the first principal component describes 99% of the overall variance, this indicates high multicollinearity among the input/independent variables. Similar to [16], instead of modeling the real response we generated four different outputs as follows

$$y_1 = \exp(\mathbf{x}^{\mathrm{T}}\mathbf{x}/2c) \tag{13}$$

$$y_2 = \exp(\mathbf{x}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{x}/2c_1) \tag{14}$$

$$y_3 = (\mathbf{x}^{\mathrm{T}}\mathbf{x}/c)^3 \exp(\mathbf{x}^{\mathrm{T}}\mathbf{x}/2c) \tag{15}$$

$$y_4 = 0.3y_1 + 0.25y_2 - 0.7y_3 \tag{16}$$

where $\mathbf{A}$ is a symmetric matrix with off-diagonal elements set to 0.8 and diagonal elements set to 1.0, and c and c_1 are averages of $\{\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_i\}_{i=1}^{80}$ and $\{\mathbf{x}_i^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{x}_i\}_{i=1}^{80}$.

The first 20 examples are used to create a training data set $\mathcal{L}$, the last 20 examples are utilized to create a testing data set, and the remaining examples form a training data set $\mathcal{U}$. In order to make the synthetic outputs (13)-(16) more realistic, Gaussian white noise with different levels is added, where noise level, denoted as n/s, corresponds to ratios of the standard deviation of the noise and the clean output variables. In this study, we set $n/s \in \{15\%, 30\%, 60\%, 90\%\}$. For each noise level, 25 different training sets are generated.

5 Experiments and Discussions

In order to assess prediction performance, average relative error (δ) and correlation coefficient (R) indices are used, which are formally defined as follows.

$$\delta = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} \tag{17}$$

$$R = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2 \sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}} \tag{18}$$

where y_i and $\hat{y}_i$ are actual and predicted outputs, respectively, and $\bar{y}$ and $\bar{\hat{y}}$ are average of actual and predicted outputs, respectively.

Here, the radial basis function (RBF) kernel function is adopted: $K(\mathbf{x}, \mathbf{z}) = \exp(-p\|\mathbf{x} - \mathbf{z}\|^2), p > 0$. The reasons are four-fold: (a) the linear kernel function is a special case of RBF [17]; (b) The Sigmoid kernel function is not positive definite, and for certain parameters, the Sigmoid kernel function behaves like RBF [18]; (c) Relatively, there are more parameters in the polynomial kernel function so that it is more difficult for model selection. In addition, the polynomial kernel function has also numerical difficulties, such as overflow or underflow; (d) The RBF kernel possesses good smoothness properties, which are usually preferred in the case one does not have a priori knowledge about the problem [19, 20].

5.1 Parameters Optimization

As is well-known, the values of parameters (γ, λ, p, k) may influence largely the performance of LS-SVR and S^{2}LS-SVR. In order to identify proper parameters, the grid search [21, 22] is used in the case of LS-SVR. Let $\gamma \in \{2^{-5}, 2^{-3}, \cdots, 2^{15}\}$ and $p \in \{2^{-15}, 2^{-13}, \cdots, 2^{3}\}$. For all possible combinations (γ, p), the average relative error δ is calculated using leave-one-out (LOO) procedure. Thus, an optimal pair γ^*, p^* can be determined. In the case of S^{2}LS-SVR, the two-hierarchical grid search is utilized. Specifically, in the first level, an optimal pair (γ^*, p^*) is selected on the training set $\mathcal{L}$ with LS-SVR. These parameters are then fixed at these values. In the second level, let $\lambda \in \{2^{-5}, 2^{-3}, \cdots, 2^{15}\}, k \in \{1, 2, \cdots, 5\}$. The remaining parameters (λ, k) are determined using a grid search and LOO procedure while introducing the training set $\mathcal{U}$. The criterion to measure prediction performance is still the average relative error δ on the training set $\mathcal{L}$. In this level, an optimal pair (λ^* and k^*) is determined.

5.2 Performance Comparison

With the optimal parameters γ^*, λ^*, p^* and k^* obtained in subsection 5.1, two regression models are built with LS-SVR and S^{2}LS-SVR, respectively. Then they are applied to predict the resulting outputs on the testing set, and the corresponding assessment indices are calculated and reported in Table 1. It is worth mentioning that the values in Table 1 correspond to average of 25 different simulations. The correlation coefficients for kernel PLS (Partial Least-Squares) regression are taken directly from [16], but average relative errors are not provided. Furthermore, kernel PLS regression model is built on the first 60 examples in corn data set.

Table 1: Comparisons of the predicted results with kernel PLS, LS-SVR and S^{2}LS-SVR

	n/s	y_1		y_2		y_3		y_4	
		δ	R	δ	R	δ	R	δ	R
LS-SVR	15%	0.0165	0.9818	0.0231	0.9708	2.6555	0.9429	1.1252	0.9098
	30%	0.0150	0.9864	0.0216	0.9842	2.5643	0.9381	1.0542	0.9345
	60%	0.0168	0.9003	2.6872	0.9306	2.6872	0.9306	1.1133	0.8802
	90%	0.0171	0.7609	2.5982	0.6784	2.5982	0.6784	1.0959	0.6762
Kernel PLS	15%	–	0.99	–	0.99	–	0.98	–	0.98
	30%	–	0.97	–	0.97	–	0.94	–	0.95
	60%	–	0.89	–	0.89	–	0.88	–	0.88
	90%	–	0.85	–	0.77	–	0.85	–	0.85
S^{2}LS-SVR	15%	0.0026	0.9946	0.0059	0.9922	1.1101	0.9627	0.5015	0.9618
	30%	0.0050	0.9943	0.0079	0.9918	1.4070	0.9581	0.5618	0.9581
	60%	0.0082	0.9941	0.0114	0.9916	1.6750	0.9599	0.6471	0.9581
	90%	0.0108	0.9941	0.0142	0.9897	1.9021	0.9615	0.7614	0.8814

From Table 1, it is not difficult to see that our S^{2}LS-SVR outperforms obviously kernel PLS regression and LS-SVR, and kernel PLS regression outperforms LS-SVR. In our opinion, there are two main reasons: (a) The S^{2}LS-SVR considers the training set $\mathcal{U}$ as well as the training set $\mathcal{L}$ when constructing the model. This indicates that the training set $\mathcal{U}$ can boost the predictive performance of the LS-SVR. Though the training set $\mathcal{U}$ is available in advance, the kernel PLS regression and LS-SVR cannot exploit the underlying knowledge in the set $\mathcal{U}$, since the resulting labels are unknown. (b) The kernel PLS regression comes next to S^{2}LS-SVR in performance. The main reason is that it considers first 60 examples in corn data set. Another way to say this is that the actual labels for training set $\mathcal{U}$ are also considered when constructing the corresponding models. In theory, the model that is built on the basis of the information is upper limit of the counterpart semi-supervised model [3].

Additionally, the performance of kernel PLS regression and LS-SVR gradually decreases as white noise level (n/s) increases. In particular, the change in the predicted performance for LS-SVR is more obvious. We think that reasons are two-fold: (a) Since the labels for training set $\mathcal{U}$ are estimated from neighbors' labels in training set $\mathcal{L}$, so that estimated labels may be closer to real ones than those contaminated by white noise. Thus, the influence that white noise on the model performance may be offset largely. (b) The kernel PLS regression represents input and

output matrix with principal components in feature space $\mathcal{H}$, so that the information underlying in input and output matrix can be made use of. Furthermore, principal components themselves can greatly reduce the effect of noise.

6 Conclusions

Motivated by semi-supervised ridge regression model, this paper proposes a novel semi-supervised method for regression problem, S^{2}LS-SVR, on the basis of the LS-SVR. Firstly, the labels of the un-annotated examples are estimated according to the nearest neighborhood characteristic in the feature space. The information is then introduced when constructing the S^{2}LS-SVR model. On the one hand, one can enrich the training set, and provide more training examples for supervised learning algorithm. On the other hand, the nearest neighborhood characteristic can reduce the effect of noise, thus improving the robustness of the model. Finally, experimental results on corn data set show the S^{2}LS-SVR outperforms the kernel PLS regression and LS-SVM, which verifies the feasibility and efficiency of the S^{2}LS-SVR method.

References

[1] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT), Madison, Wisconsin, United States, 1998, 92-100

[2] X. Zhu, Semi-supervised Learning Literature Survey, Technical Report 1530, Department of Computer Sciences, University of Wisconsin, Madison, 2008

[3] O. Chapelle, B. Schölkopf, A. Zien, Semi-supervised Learning, MIT Press, Cambridge, 2006

[4] X. Liu, Z. Hao, X. Yang, X. Deng, Robustness of semi-supervised learning algorithm LLGC trained using soft labels for misclassified data, Journal of Information & Computational Science, 7 (2010), 1827-1837

[5] O. Chapelle, V. Sindhwani, S. Keerthi, Optimization techniques for semi-supervised support vector machines, Journal of Machine Learning Research, 9 (2008), 203-233

[6] C. Cortes, M. Mohri, On transductive regression, Advances in Neural Information Processing Systems 19, B. Schölkopf, J. Platt, T. Hoffman, editors, MIT Press, Cambridge, MA, 2007, 305-312

[7] U. Brefeld, T. Cärtner, T. Scheffer, S. Wrobel, Efficient co-regularised least squares regression, Proceedings of the 23nd International Conference on Machine Learning (ICML), 2006, 137-144

[8] Z. H. Zhou, M. Li, Semisupervised regression with cotraining-style algorithms, IEEE Transactions on Knowledge and Data Engineering, 19 (2007), 1479-1493

[9] Z. H. Zhou, M. Li, Semi-supervised regression with co-training, Proceedings of the 19th International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, 2005, 908-913

[10] V. N. Vapnik, The Nature of Statistical Learning Theory, 2nd Edition, Springer Verlag, New York, 1999

[11] V. N. Vapnik, Statistical learning Theory, John Wiley & Sons, Inc., New York, 1998

[12] J. A. K. Suyken, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, Least Squares Support Vector Machines, World Scientific Pub. Co., Singapore, 2002

[13] T. Van Gestel, J. A. K. Suykens, B. Baesens, S. Viaene, J. Vanthienen, G. Dedene, B. De Moor, J. Vandewalle, Benchmarking least squares support vector machine classifiers, Machine Learning, 54 (2004), 5-32

[14] X. An, S. Xu, L. D. Zhang, S. G. Su, Multiple dependent variable LS-SVM regression algorithm and its application in NIR spectral quantitative analysis, Spectroscopy and Spectral Analysis, 29 (2009), 127-130 (in Chinese)

[15] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, Cambridge, 2004

[16] R. Rosipal, L. J. Trejo, Kernel Partial least squares regression in reproducing kernel Hilbert space, Journal of Machine Learning Research, 2 (2001), 97-123

[17] S. S. Keerthi, C. J. Lin, Asymptotic behaviros of support vector machines with Gaussian kernel, Neural Computation, 15 (2003), 1667-1689

[18] H. T. Lin, C. J. Lin, A Study on Sigmoid Kernels for SVM and the Training of non-PSD Kernels by SMO-Type Methods, Technical Report, Department of Computer Science, National Taiwan University, 2003

[19] A. J. Smola, B. Schölkopf, K.-R. Müller, The connection between regularization operators and support vector kernels, Nueral Networks, 11 (1998), 637-649

[20] F. Girosi, An equivalence between sparse approximation and support vector machines, Neural Computation, 10 (1998), 1455-1480

[21] C. W. Hsu, C. C. Chang, C. J. Lin, A practical guide to support vector classification. Available [online]: http://www.csie.ntu.edu.tw/ cjlin/ papers//guide/guide.pdf

[22] S. Xu, F. J. Ma, L. Tao, Learn from the information contained in the false splice sites as well as in the true splice sites using SVM. Proceedings of the Interantional Conference on Intelligent Systems and Knowledge Engineering (ISKE), Chengdu, China, 2007, 1360-1366