[8] B. Samuelsson and C. Troein, "Superpolynomial growth in the number of attractors in Kauffman networks," *Phys. Rev. Lett.*, vol. 90, no. 9, pp. 098701-1–098701-4, Mar. 2003.
[9] Q. Zhao, "A remark on 'Scalar equations for synchronous Boolean networks with biological applications' by C. Farrow, J. Heidel, J. Maloney, and J. Rogers," *IEEE Trans. Neural Netw.*, vol. 16, no. 6, pp. 1715–1716, Nov. 2005.
[10] D. Cheng, Z. Li, and H. Qi, "Realization of Boolean control networks," *Automatica*, vol. 46, no. 1, pp. 62–69, Jan. 2010.
[11] D. Cheng, H. Qi, Z. Li, and J. Liu, "Stability and stabilization of Boolean networks," *Int. J. Robust Nonlinear Control*, vol. 21, no. 2, pp. 134–156, Jan. 2011.
[12] D. Cheng and H. Qi, "Controllability and observability of Boolean control networks," *Automatica*, vol. 45, no. 7, pp. 1659–1667, Jul. 2009.
[13] D. Cheng and H. Qi, "State–space analysis of Boolean networks," *IEEE Trans. Neural Netw.*, vol. 21, no. 4, pp. 584–594, Apr. 2010.
[14] M. Anguelova and B. Wennberg, "On analytic and algebraic observability of nonlinear delay systems," *Automatica*, vol. 46, no. 4, pp. 682–686, Apr. 2010.
[15] L. Guo and Z. Wang, "Exact boundary observability for nonautonomous quasilinear wave equations," *J. Math. Anal. Appl.*, vol. 364, no. 1, pp. 41–50, Apr. 2010.
[16] S. Zhao and J. Sun, "Controllability and observability for impulsive systems in complex fields," *Nonlinear Anal.: Real World Appl.*, vol. 11, no. 3, pp. 1513–1521, Jun. 2010.
[17] Z. Wang, J. Lam, G. Wei, K. Fraser, and X. Liu, "Filtering for nonlinear genetic regulatory networks with stochastic disturbances," *IEEE Trans. Autom. Control*, vol. 53, no. 10, pp. 2448–2457, Nov. 2008.
[18] W. Yu, J. Lu, G. Chen, Z. Duan, and Q. Zhou, "Estimating uncertain delayed genetic regulatory networks: An adaptive filtering approach," *IEEE Trans. Autom. Control*, vol. 54, no. 4, pp. 892–897, Apr. 2009.

Feature Selection Using Probabilistic Prediction of Support Vector Regression

Jian-Bo Yang and Chong-Jin Ong

***Abstract*— This brief presents a new wrapper-based feature selection method for support vector regression (SVR) using its probabilistic predictions. The method computes the importance of a feature by aggregating the difference, over the feature space, of the conditional density functions of the SVR prediction with and without the feature. As the exact computation of this importance measure is expensive, two approximations are proposed. The effectiveness of the measure using these approximations, in comparison to several other existing feature selection methods for SVR, is evaluated on both artificial and real-world problems. The result of the experiments show that the proposed method generally performs better than, or at least as well as, the existing methods, with notable advantage when the dataset is sparse.**

***Index Terms*— Feature ranking, feature selection, probabilistic predictions, random permutation, support vector regression.**

I. Introduction

Feature selection plays an important role in pattern recognition, data mining, and information retrieval and has been the subject of intense research in the past decade. Generally, methods for feature selection can be classified into two categories: filter and wrapper methods [1], [2]. Wrapper methods rely heavily on the specific structure of the underlying learning algorithm, whereas filter methods are independent of it. Due to its more involved nature, wrapper methods usually yield better performance than filter methods but have a heavier computational load.

With a few exceptions [3]–[6], most feature selection methods are developed for use in classification problems. One possible reason for this is the ease of formulation of criteria for feature selection by exploiting the discriminability of classes. While some methods can be extended from classification to regression applications [3], [7], others may not. Straightforward adaptation by discretizing (or binning) the target variable into several classes is not always desirable, as substantial loss of important ordinal information may result.

This brief proposes a new wrapper-based feature selection method for support vector regression (SVR), motivated by our earlier work on classification problem using support vector machine (SVM) [8] and multilayer perceptron neural networks [9]. Under the probabilistic framework, the output of a standard SVR can be interpreted as $p(y|x)$, the conditional density function of target $y \in R$ given input $x \in R^d$ for a given dataset. The proposed method relies on the sensitivity of $p(y|x)$ with respect to a given feature as a measure of importance of this feature. More exactly, the importance score of a feature is the aggregation over the feature space of the difference of $p(y|x)$ with and without the feature. The exact computation of the proposed method is expensive, so two approximations are proposed. Each of the approximations, embedded in an overall feature selection scheme, is tested on various artificial and real-world datasets and compared with several other existing feature selection methods. Experimental results show that the proposed method performs generally better than, if not at least as well as, other methods in almost all experiments.

This brief is organized as follows. Section II reviews the formulation of probabilistic SVR and other relevant information. Details of the proposed feature ranking criterion and two approximations are presented in Section III. Section IV shows the overall feature selection scheme. Results of numerical experiments of the proposed method, benchmarked against other methods, are reported in Section V. Section VI concludes this brief.

II. Review of Past Works

Standard SVR [10] obtains the regressor function $f(x) := \omega'\phi(x)+b$ for a dataset $\mathcal{D} := \{(x_i, y_i) : i \in \mathcal{I}_\mathcal{D}\}$ with $x_i \in R^d$ and $y \in R$ by solving the following primal problem (PP) over ω, b, ξ, ξ^*:

$$\min_{\omega,\xi_i\geq 0,\xi_i^*\geq 0} \quad \frac{1}{2}\omega'\omega + C\sum_{i\in\mathcal{I}_\mathcal{D}}\left(\xi_i+\xi_i^*\right) \tag{1}$$

$$\text{s.t.} \quad y_i - \omega'\phi(x_i) - b \leq \epsilon + \xi_i \quad \forall i \in \mathcal{I}_\mathcal{D} \tag{2}$$

$$\omega'\phi(x_i) + b - y_i \leq \epsilon + \xi_i^* \quad \forall i \in \mathcal{I}_\mathcal{D}. \tag{3}$$

Manuscript received February 24, 2010; revised November 21, 2010; accepted March 6, 2011. Date of publication May 5, 2011; date of current version June 2, 2011.

The authors are with the Department of Mechanical Engineering, National University of Singapore, 117576, Singapore (e-mail: yangjianbo@nus.edu.sg; mpeongcj@nus.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2011.2128342

The function $\phi : R^d \to \mathcal{H}$ maps x into a high-dimensional Hilbert space $\mathcal{H}$, and $\omega \in \mathcal{H}$, $b \in R$ are variables that define $f(x)$ with ξ_i, ξ_i^* being the nonnegative slack variables needed to enforce constraints (2) and (3). The regularization parameter, i.e., $C > 0$, trades offs the size of ω and the amount of slack, s while the parameter $\epsilon > 0$ specifies the allowable deviation of the $f(x_i)$ from y_i. In practice, PP is often solved through its dual problem (DP)

$$\max_{\alpha,\alpha^*} \quad -\frac{1}{2}\sum_{i\in\mathcal{I}_\mathcal{D}}\sum_{j\in\mathcal{I}_\mathcal{D}}\left(\alpha_i-\alpha_i^*\right)\left(\alpha_j-\alpha_j^*\right)K\left(x_i,x_j\right) - \epsilon\sum_{i\in\mathcal{I}_\mathcal{D}}\left(\alpha_i+\alpha_i^*\right)+\sum_{i\in\mathcal{I}_\mathcal{D}} y_i\left(\alpha_i-\alpha_i^*\right) \tag{4a}$$

$$\sum_{i\in\mathcal{I}_\mathcal{D}}\left(\alpha_i-\alpha_i^*\right)=0,\ 0\le\alpha_i\le C,\ 0\le\alpha_i^*\le C,\ i\in\mathcal{I}_\mathcal{D} \tag{4b}$$

where α_i and α_i^* are the respective Lagrange multipliers of (2) and (3), $\omega = \sum_{i\in\mathcal{I}_\mathcal{D}}(\alpha_i - \alpha_i^*)\phi(x_i)$, and $K(x_i, x_j) = \phi(x_i)'\phi(x_j)$. Using these expressions, the regressor function is known to be

$$f(x) = \omega'\phi(x_i) + b = \sum_{i\in\mathcal{I}_\mathcal{D}}\left(\alpha_i-\alpha_i^*\right)K(x_i,x)+b. \tag{5}$$

Expression (5) provides an estimate $f(x)$ for output y for any x but provides no information on the confidence level of this estimate. Recognizing this shortcoming, several attempts to incorporate probabilistic values to SVR output has been reported in the literature. Following the approach of Bayesian framework for neural network [11], Law and Kwok [12] proposed a Bayesian SVR (BSVR) formulation incorporating probabilistic information. Gao *et al.* [13] improved upon BSVR by deriving the evidence and error bar approximation. Chu *et al.* [14] proposed the use of a unified loss function over the standard ϵ-insensitive loss function and provided better accuracy in evidence evaluation and inferences. Lin and Weng [15] follow the neural network [16] approach by assuming that a deterministic regressor model exists and the SVR is an attempt to represent this model. In this setting, the output y is modeled as the SVR regressor function with an additive noise in the form of

$$y = f(x) + \zeta \tag{6}$$

where ζ belongs to the Laplace or the Gaussian distributions. It is then possible to assume that the SVR output corresponds to the conditional density function of $p(y|x)$. With (6), this means that density functions of y for a given x are

$$p^L(y|x;\sigma) = \frac{1}{2\sigma}\exp\left(-\frac{|y-f(x)|}{\sigma}\right) \tag{7}$$

$$p^G(y|x;\sigma) = \frac{1}{\sqrt{2\pi}\,\sigma}\exp\left(-\frac{(y-f(x))^2}{2\sigma^2}\right) \tag{8}$$

for the Laplace and Gaussian cases, respectively. Like the neural network approach, the intention is to obtain estimates of σ of (7) and (8) from $\mathcal{D}$. If $p(x, y)$ is the joint density function of x and y, the likelihood function, as a function of σ, of observing $\mathcal{D}$ is given by

$$L(\sigma) = \Pi_{i\in\mathcal{I}_\mathcal{D}}\, p(x_i, y_i) = \Pi_{i\in\mathcal{I}_\mathcal{D}}\, p(y_i|x_i;\sigma)\, p(x_i)$$

under the assumption of independent and identically distributed samples. By further assuming that $p(x)$ is independent of σ, the expressions of σ can be obtained by maximizing the logarithm function of $L(\sigma)$ [16]. These expressions are

$$\sigma^L = \frac{\sum_{i\in\mathcal{I}_\mathcal{D}}|y_i - f(x_i)|}{|\mathcal{I}_\mathcal{D}|} \tag{9}$$

$$\left(\sigma^G\right)^2 = \frac{\sum_{i\in\mathcal{I}_\mathcal{D}}(y_i - f(x_i))^2}{|\mathcal{I}_\mathcal{D}|} \tag{10}$$

for the Laplace and Gaussian distributions, respectively. It has been shown [15] that this approach is competitive in terms of performance to the BSVR methods. In view of this, the proposed feature selection method uses this approach and relies on (7) and (8) for its computation.

III. Proposed Feature Selection Criterion for Regression

The proposed method of selection of feature importance relies on measures of the difference between two density functions. Our choice of this measure is the well-known Kullback–Leibler divergence (KL divergence) $D_{KL}(\cdot;\cdot)$. The use of KL divergence has appeared in the past (for example [17], [18] and reference therein) as a filter feature selection method. Typically, these methods look for the feature that maximizes the KL distance between $p(x^j)$ (or $p(y|x^j)$) and $p(y)$ although other variations exist. Unlike theirs, we use this to measure the difference of two density functions in a wrapper method for the SVR problem. Given two distributions $p(y)$ and $q(y)$

$$D_{KL}(p(y);q(y)) = \int p(y)\log\frac{p(y)}{q(y)}dy. \tag{11}$$

From its definition, it is easy to verify that $D_{KL}(p(y); q(y)) \ge 0$ for any $p(y)$ and $q(y)$, $D_{KL}(p(y); q(y)) = 0$ if and only if $p(y) = q(y)$ and $D_{KL}(p(y); q(y))$ is not symmetrical with respect to its arguments. The last property is a result of treating $p(y)$ as the reference distribution.

In the case of SVR, the density function $p(y|x)$ at any x is assumed to be (7) or (8) with $f(\cdot)$ being the solution obtained from (5). Given $x \in R^d$, $x_{-j} \in R^{d-1}$ can be obtained by removing the jth feature from x or, equivalently, $x_{-j} = Z_j^d x$ where Z_j^d is the $(d-1)\times d$ matrix obtained by removing the jth row of the $d \times d$ identity matrix. With this, the difference of the two density functions $p(y|x)$ and $p(y|x_{-j})$ at a particular x (and hence x_{-j}) is $D_{KL}(p(y|x); p(y|x_{-j}))$. The proposed feature importance measure is an aggregation of $D_{KL}(p(y|x); p(y|x_{-j}))$ over all x in the x space. More exactly, the measure is

$$S_D(j) = \int D_{KL}\left(p(y|x);\, p\left(y|x_{-j}\right)\right)p(x)\,dx. \tag{12}$$

The motivation for defining S_D is simple: the greater the D_{KL} divergence between $p(y|x)$ and $p(y|x_{-j})$ over the x space, the greater the importance of the jth feature. For convenience, (12) is termed the sensitivity of density functions or SD.

In (12), $p(y|x)$ is either (7) or (8) with $f(\cdot)$ of (5) trained on $\mathcal{D}$. Similarly, $p(y|x_{-j})$ is obtained from the SVR output function trained using the derived dataset $\mathcal{D}_{-j} = \{(x_{-j,i}, y_i) : i \in \mathcal{I}_{\mathcal{D}}\}$, where $x_{-j,i} \in R^{d-1}$ is the ith sample of the derived vector x_{-j}. Thus, evaluations of $S_D(j)$, $j = 1, \ldots, d$ require the training of SVR d times, each with $\mathcal{D}_{-j}$ for a different j. Clearly, this process is computationally expensive. Following [8], a random permutation (RP) or scrambling process [19] is used to approximate $p(y|x_{-j})$ such that the retraining of SVR is avoided. The basic idea of the RP process is to scramble the values of the jth feature in $\mathcal{D}$ while keeping the values of all other features unchanged. Specifically, let x_i^j be the value of the j feature of sample i and $\{\eta_1, \ldots, \eta_n\}$ be a set of numbers drawn from a discrete uniform distribution in the interval from 1 to n. Then, for each i starting from 1 to n, swap the values of x_i^j and $x_{\eta_i}^j$.

Let $x_{(j)} \in R^d$ be the sample derived from x after the RP process on the jth feature and let $p(y|x_{(j)})$ be the conditional density function of y given $x_{(j)}$.

Theorem 1:

$$p\left(y|x_{(j)}\right) = p\left(y|x_{-j}\right). \tag{13}$$

The proof of this theorem is given in [8]. The theorem is stated for the case where the exact $p(y|x)$, $p(y|x_{(j)})$ and $p(y|x_{-j})$ are known. In the case where they are approximated from a dataset, the equality of (13) becomes an approximation. Nevertheless, our experiment shows that the approximation is very good, even when the data is sparse.

The utility of Theorem 1 is clear. The density function $p(y|x_{-j})$ of (12) can be replaced by $p(y|x_{(j)})$. Such a replacement brings about significant computational advantage. By assuming that $p(y|x_{(j)})$ can be evaluated from (7) or (8) using $f(x_{(j)})$ obtained from the SVR training using $\mathcal{D}$ (since x and $x_{(j)}$ are both d-dimensional), this avoids the expensive d-time retraining of SVR on $\mathcal{D}_{-j}$. Correspondingly, (12) can be equivalently stated as

$$S_D(j) = \int D_{KL}\left(p(y|x); p\left(y|x_{(j)}\right)\right) p(x)\, dx. \tag{14}$$

Fig. 1 shows a plot of $p(y_i|x_i)$ and $p(y_i|x_{(j),i})$ at one choice of x_i for a typical SVR problem with $d = 1$. To compute the S_D, further approximation of (14) is needed, resulting in

$$\hat{S}_D(j) = \frac{1}{|\mathcal{I}_{\mathcal{D}}|} \sum_{i \in \mathcal{I}_{\mathcal{D}}} D_{KL}\left(p(y_i|x_i); p\left(y_i|x_{(j),i}\right)\right). \tag{15}$$

When $p(y|x)$ and $p(y|x_{(j)})$ are Laplace functions or Gaussian functions, explicit expressions of $\hat{S}_D(j)$ exist. Using (7), the KL divergence for the case of Laplace function can be shown to be

$$\begin{aligned} &D_{KL}\left(p^L\left(y|x;\sigma^L\right); p^L\left(y|x_{(j)};\sigma_{(j)}^L\right)\right) \\ &= \ln\frac{\sigma_{(j)}^L}{\sigma^L} - 1 + \frac{\sigma^L}{\sigma_{(j)}^L}\exp\left(-\frac{|f(x) - f\left(x_{(j)}\right)|}{\sigma^L}\right) \\ &\quad + \frac{|f(x) - f\left(x_{(j)}\right)|}{\sigma_{(j)}^L} \end{aligned} \tag{16}$$

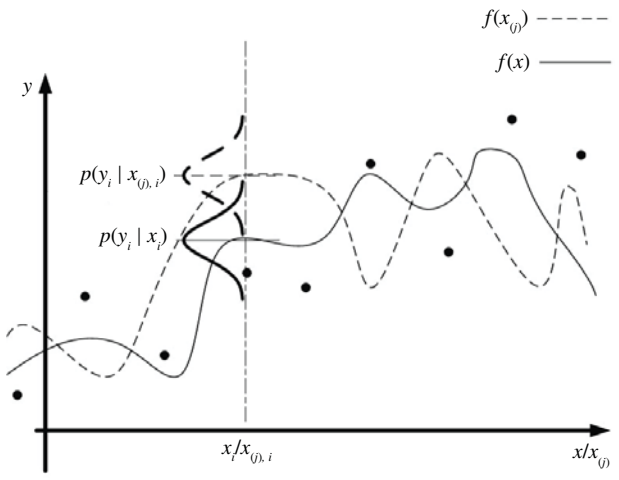

Fig. 1. Demonstration of the proposed feature ranking criterion with $d = 1$. Dots indicate locations of y_i.

for a given x where σ^L is given by (9) and $\sigma_{(j)}^L$ is obtained from (9) by replacing $f(x)$ with $f(x_{(j)})$. Using (16) in (15) and removing associated constants yield

$$\begin{aligned} \hat{S}_D^L(j) = \frac{1}{|\mathcal{I}_{\mathcal{D}}|}\sum_{i \in \mathcal{I}_{\mathcal{D}}}\left[\frac{\sigma^L}{\sigma_{(j)}^L}\exp\left(-\frac{|f(x_i) - f\left(x_{(j),i}\right)|}{\sigma^L}\right)\right. \\ \left. + \frac{|f(x_i) - f\left(x_{(j),i}\right)|}{\sigma_{(j)}^L} + \ln\frac{\sigma_{(j)}^L}{\sigma^L}\right]. \end{aligned} \tag{17}$$

Following the same development for the case when $p(y|x)$ is Gaussian, the expressions are:

$$\begin{aligned} &D_{KL}\left(p^G\left(y|x;\sigma^G\right); p^G\left(y|x_{(j)};\sigma_{(j)}^G\right)\right) = \ln\frac{\sigma_{(j)}^G}{\sigma^G} \\ &+ \frac{f(x)^2 + f\left(x_{(j)}\right)^2 + \left(\sigma^G\right)^2 - 2f(x) f\left(x_{(j)}\right)}{2\left(\sigma_{(j)}^G\right)^2} - \frac{1}{2} \end{aligned} \tag{18}$$

$$\begin{aligned} \hat{S}_D^G(j) = \frac{1}{2|\mathcal{I}_{\mathcal{D}}|}\sum_{i \in \mathcal{I}_{\mathcal{D}}}\left[\frac{\left(f(x_i) - f\left(x_{(j),i}\right)\right)^2}{\left(\sigma_{(j)}^G\right)^2}\right. \\ \left. + \left(\frac{\sigma^G}{\sigma_{(j)}^G}\right)^2 + 2\ln\frac{\sigma_{(j)}^G}{\sigma^G}\right] \end{aligned} \tag{19}$$

where the expression of (18) is given by [20].

In summary, $\hat{S}_D(j)$ can be computed for all $j = 1, \ldots, d$, after a one-time training of SVR, one-time evaluation of σ^L (or σ^G), d-time RP process, d-time evaluation of $\sigma_{(j)}^L$ (or $\sigma_{(j)}^G$), and d-time evaluation of D_{KL}.

Remark 1: The kernel matrix is different for each of the d-time evaluation of $\sigma_{(j)}^L$ (or $\sigma_{(j)}^G$) and this incurs additional computations. Such computations can be kept low using update formulae. Suppose x_r, x_q and $x_{(j),r}, x_{(j),q}$ are two samples before and after the RP process is applied to feature j. It is easy to show that $K(x_{(j),r}, x_{(j),q}) = K(x_r, x_q) + x_{(j),r}^j * x_{(j),q}^j - x_r^j * x_q^j$ for linear kernel and $K(x_{(j),r}, x_{(j),q}) = K(x_r, x_q) * \exp[\kappa(x_r^j - x_q^j)^2 - \kappa(x_{(j),r}^j - x_{(j),q}^j)^2]$ with kernel parameter κ for Gaussian kernel.

IV. Feature Selection Scheme

The proposed $\hat{S}_D^L$ and $\hat{S}_D^G$ can be used in two ways. The most obvious is when it is used once to yield a ranking list of all features based on a one-time training of SVR on $\mathcal{D}$. It can also be used for more extensive ranking schemes like the recursive feature elimination (RFE) scheme. Basically, the RFE approach works in iterations. In each iteration, a ranking of all remaining features is obtained using some appropriate measures ($\hat{S}_D^L$, $\hat{S}_D^G$ or others). The least important feature, as determined by the measure, is then removed from further consideration. This procedure stops after $n-r$ iterations to yield the top r features. Accordingly, the overall scheme with respect to measure $\hat{S}_D^L$ ($\hat{S}_D^G$) is referred to as SD-L-RFE (SD-G-RFE). Inputs to scheme SD-L-RFE are $\mathcal{D}$ and $\Gamma = \{1, \ldots, d\}$, while the output is a ranked list of features in the form of an index set $\Gamma^\dagger = \{\gamma_1^\dagger, \ldots, \gamma_d^\dagger\}$ where $\gamma_j^\dagger \in \Gamma$ for each $j = 1, \ldots, d$ in decreasing order of importance.

Following Theorem 1, the associated computational costs of the SD-L-RFE (SD-G-RFE) scheme is the training of SVR at each iteration and the evaluations of $\hat{S}_D^G(j)(\hat{S}_D^L(j))$ using (18) (16) for each j of the remaining features in that iteration. This is the case of the proposed scheme. In the next section where other benchmark methods are discussed, the retraining of SVR at each iteration and within the iteration may be needed for the ranking of features because of the inapplicability of Theorem 1.

V. Experiment

This section presents the result of a numerical experiment of SD-L-RFE, SD-G-RFE, and the following five existing benchmark methods on artificial and real-world data sets.

1) Mutual information (MI) method [18]: It measures the importance of a feature by considering both the MI between this feature and target and the MI between this feature and the selected ones.
2) Dependence maximization method [4]: It uses cross-covariance in the kernel space, known as the Hilbert–Schmidt norm of cross-covariance operator (HSIC) [21], as a dependence measure between feature variables and target variable. The importance of a feature is measured by its sensitivity to this dependence measure. This method is used because of its relative good performances despite its known limitations [22].
3) SVM-RFE method ($\Delta\|\omega\|^2$) [3], [5]: It measures the importance of a feature by the sensitivity of the cost function (1) with and without this feature.
4) SVR radius-margin bound method (RMB) [5]: It measures the importance of a feature by its sensitivity w.r.t. SVR RMB.
5) SVR span bound method (SpanB) [5]: It measures the importance of a feature by its sensitivity w.r.t. SVR SpanB bound.

The first two benchmark methods are filter methods, while the last three are wrapper methods. All methods, except the MI method, use the same RFE scheme described in Section IV for ranking the features, and hence they are referred to as mRMR, HSIC-RFE, $\Delta\|\omega\|^2$-RFE, RMB-RFE, and SpanB-RFE, respectively.

Note that the retraining of SVR within each RFE iteration is not needed for $\Delta\|\omega\|^2$-RFE. However, in the implementation of RMB-RFE and SpanB-RFE by [5], retraining is used within each iteration of the RFE scheme. Obviously, this is much more expensive than the proposed method since the result of Theorem 1 is not applicable to them. Our experiments include both cases: RMB-RFE and SpanB-RFE when retraining is not used and RMB-RFE* and SpanB-RFE* when it is.

For each experiment dataset, the result is reported over 30 realizations, which are created by random (stratified) sampling of the set $\mathcal{D}$ into subsets $\mathcal{D}_{trn}$ and $\mathcal{D}_{tst}$. As usual, $\mathcal{D}_{trn}$ is used for SVR training, hyper-parameter tuning, and feature ranking, while $\mathcal{D}_{tst}$ is used for unbiased evaluation of the feature selection performance. For each realization, $\mathcal{D}_{trn}$ is normalized to zero mean and unit standard deviation, and its normalization parameters are then used to normalize $\mathcal{D}_{tst}$. The kernel function used for all problems is the Gaussian kernel. In each experiment, all hyperparameters (C, κ, ϵ) are chosen by a fivefold cross validation on the first five realizations of $\mathcal{D}_{trn}$, and the hyperparameter corresponding to the lowest average cross-validation error among five realizations is chosen. The grid over the (C, κ, ϵ) is $[2^{-2}, 2^{-1}, \ldots, 2^6] \times [2^{-6}, 2^{-5}, \ldots, 2^2] \times [2^{-5}, 2^{-4}, \ldots, 2^2]$.

Two well-known regression performance measures, i.e., mean squared error (MSE) and squared correlation coefficient (SCC), are used to evaluate the performance. They are given by

$$\text{MSE} := \frac{\sum_{i=1}^{|\mathcal{D}_{tst}|} \left(\hat{y}_i - y_i\right)^2}{|\mathcal{D}_{tst}|}$$

$$\text{SCC} := \frac{\left(|\mathcal{D}_{tst}| \sum_i \hat{y}_i y_i - \sum_i \hat{y}_i \sum_i y_i\right)^2}{\left(|\mathcal{D}_{tst}| \sum_i \hat{y}_i^2 - \sum_i \hat{y}_i \sum_i \hat{y}_i\right)\left(|\mathcal{D}_{tst}| \sum_i y_i^2 - \sum_i y_i \sum_i y_i\right)}$$

where y_i and $\hat{y}_i$, for $i \in \{1, \ldots, |\mathcal{D}_{tst}|\}$, are the true and predicted target values, respectively.

Statistical paired t-test using MSE and SCC are conducted for all problems. Specifically, paired t-test between SD-L-RFE and each of the other methods is conducted using different numbers of top ranked features. Herein, the null hypothesis is that the mean MSE or SCC of the two tested methods is the same against the alternate hypothesis that they are not. The chance that this null hypothesis is true is measured by the returned p-value and the significance level is set at 0.05 for all experiments. The symbols "+" and "−" are used to indicate the win or loss situation of SD-L-RFE over the other tested method.

In all experiments, the numerical algorithm for training of SVR is implemented by the LIBSVM package [23], where sequential minimal optimization method is used to solve the DP (4).

A. Artificial Problems

In this subsection, three artificial regression problems are used to evaluate the performance of every feature selection method. The first two problems were used in [24] and the last one is new. Each problem has 10 variables $x^1, \ldots, x^{10}$ and

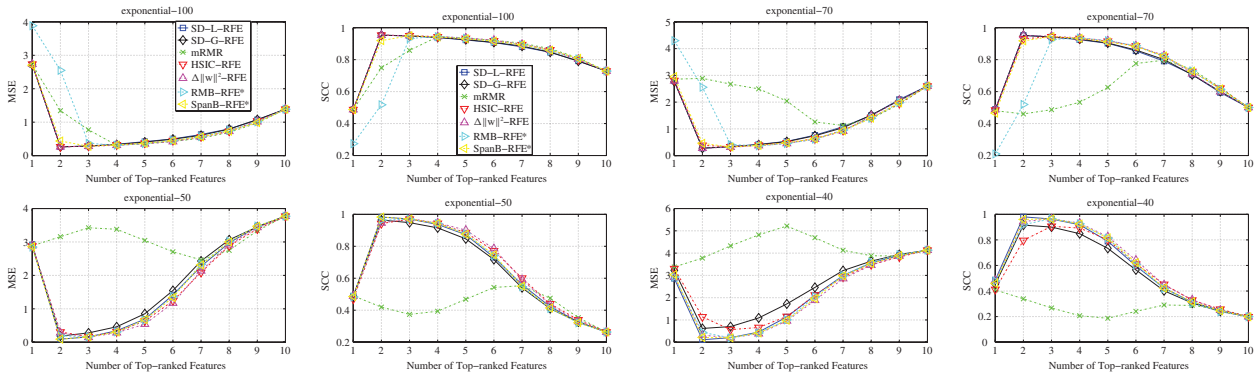

Fig. 2. Average MSE (left-hand side) and average SCC (right-hand side) against top-ranked features over 30 realizations for the exponential function problem with six different settings.

TABLE I

NUMBER OF REALIZATIONS THAT RELEVANT FEATURE IS SUCCESSFULLY RANKED IN THE TOP POSITIONS OVER 30 REALIZATIONS FOR THREE ARTIFICIAL PROBLEMS. THE BEST PERFORMANCE FOR EACH $|\mathcal{D}_{trn}|$ IS HIGHLIGHTED IN SHOWN IN BOLD

	Method\$|\mathcal{D}_{trn}|$	200	100	70	50
Additive	SD-L-RFE	**30**	27	21	**19**
	SD-G-RFE	**30**	**28**	**23**	**19**
	mRMR	19	7	1	0
	HSIC-RFE	14	5	5	3
	$\Delta\|\omega\|^2$-RFE	4	5	11	4
	RMB-RFE	0	0	0	0
	SpanB-RFE	0	1	0	0
	RMB-RFE*	**30**	25	22	9
	SpanB-RFE*	**30**	23	20	9
	Method\$|\mathcal{D}_{trn}|$	200	100	70	50
Interactive	SD-L-RFE	**30**	**30**	29	12
	SD-G-RFE	**30**	**30**	**30**	11
	mRMR	9	2	0	0
	HSIC-RFE	7	9	8	6
	$\Delta\|\omega\|^2$-RFE	0	14	9	10
	RMB-RFE	0	0	0	0
	SpanB-RFE	0	0	0	0
	RMB-RFE*	**30**	**30**	**30**	**20**
	SpanB-RFE*	**30**	**30**	**30**	16
	Method\$|\mathcal{D}_{trn}|$	100	70	50	40
Exponential	SD-L-RFE	**30**	**30**	**30**	**30**
	SD-G-RFE	**30**	**30**	29	28
	mRMR	18	2	0	0
	HSIC-RFE	**30**	29	28	22
	$\Delta\|\omega\|^2$-RFE	**30**	**30**	28	28
	RMB-RFE	0	0	0	0
	SpanB-RFE	0	1	0	1
	RMB-RFE*	4	5	29	27
	SpanB-RFE*	28	28	**30**	29

the target variable y depends on some of the features as given in their underlying functions:

1) additive function problem

$$y = 0.1\exp\left(4x^1\right) + \frac{4}{1+\exp\left(-20\left(x^2-0.5\right)\right)} + 3x^3 + 2x^4 + x^5 + \delta;$$

2) interactive function problem

$$y = 10\sin\left(\pi x^1 x^2\right) + 20\left(x^3 - 0.5\right) + 10x^4 + 5x^5 + \delta;$$

TABLE II

DESCRIPTION OF REAL-WORLD DATASETS. $|\mathcal{D}_{trn}|$, $|\mathcal{D}_{tst}|$, d, C, κ, AND ϵ REFER TO THE NUMBER OF TRAINING SAMPLES, NUMBER OF TEST SAMPLES, NUMBER OF FEATURES, AND SVR HYPERPARAMETERS C, κ, AND ϵ, RESPECTIVELY

Datasets	$\|\mathcal{D}_{trn}\|$	$\|\mathcal{D}_{tst}\|$	d	C	κ	ϵ
Mpg	353	39	7	2^6	2^{-4}	2
Abalone	1254	2923	8	2^6	2^{-5}	2
Cpusmall	820	7372	12	2^6	2^{-5}	2
Housing	456	50	13	2^6	2^{-4}	2
Bodyfat	227	25	14	2^{-2}	2^{-6}	2^{-5}
Triazines	168	18	60	2^{-1}	2^{-6}	2^{-3}

3) exponential function problem

$$y = 10\exp\left(-((x^1)^2 + (x^2)^2)\right) + \delta;$$

where $x^j,\ \forall j = 1,\ldots,10$ is uniformly distributed within the range [0, 1] for the first two problems and [–1, 1] for the last. Gaussian noise $\delta \sim \mathcal{N}(0, 0.1)$ for the first two problems, while $\delta \sim \mathcal{N}(0, 0.2)$ for the last.

Each artificial problem has 2000 samples. They are randomly split into $\mathcal{D}_{trn}$ and $\mathcal{D}_{tst}$ in the ratio of $|\mathcal{D}_{trn}|$:$|\mathcal{D}_{tst}| =$ 1:9. To investigate the effect of sparseness of the training set, decreasing sizes of $|\mathcal{D}_{trn}|$ are also used while $|\mathcal{D}_{tst}|$ is maintained at 1800.

Table I presents the number of realizations (out of 30 realizations) that relevant feature are successfully ranked as the top features by the various methods for the different settings of $|\mathcal{D}_{trn}|$. The best performance in each setting is highlighted in bold. From this table, the advantage of the proposed methods is clear. They generally performs as least as well as, if not better than, all other benchmark methods except when $|\mathcal{D}_{trn}| = 50$ in the interactive problem. For benchmark methods RMB-RFE* and SpanB-RFE*, the proposed methods yield comparable performance. It is also evident that, as the size of $|\mathcal{D}_{trn}|$ decreases, the performance of proposed methods generally degrades less than that of benchmark methods. In fact, SD-L-RFE correctly ranks the important features in the top two positions for all settings for the exponential function problem.

Fig. 2 shows the average MSE and SCC against top-ranked features over 30 realizations on $\mathcal{D}_{tst}$ for the exponential problem. Methods RMB-RFE and Span-RFE are not shown since they completely fail as shown in Table I. From this figure, the

TABLE III

t-TEST ON REAL-WORLD DATASET. p-VALUES LESS THAN 0.05 ARE HIGHLIGHTED IN BOLD. N IS THE NUMBER OF TOP RANKED FEATURES

		SD-L-RFE	SD-G-RFE		mRMR		HSIC-RFE		$\Delta\|\omega\|^2$-RFE		RMB-RFE		SpanB-RFE		RMB-RFE*		SpanB-RFE*	
Dataset	N	Mean value	Mean value	$p-$ value	Mean value	$p-$ value	Mean value	$p-$ value	Mean value	$p-$ value	Mean value	$p-$ value	Mean value	$p-$ value	Mean value	$p-$ value	Mean value	$p-$ value
MSE measure																		
	1	16.47	16.47	1.00	16.86	0.75	22.45	**0.00**+	16.47	1.00	22.45	**0.00**+	31.79	**0.00**+	22.21	**0.00**+	16.97	0.69
	2	7.71	7.71	1.00	16.32	**0.00**+	18.06	**0.00**+	7.71	1.00	17.77	**0.00**+	18.35	**0.00**+	17.75	**0.00**+	8.59	0.25
	3	6.76	6.76	1.00	15.51	**0.00**+	15.67	**0.00**+	7.54	0.22	17.39	**0.00**+	16.29	**0.00**+	17.31	**0.00**+	7.69	0.15
mpg	4	6.81	6.81	1.00	13.46	**0.00**+	13.46	**0.00**+	6.88	0.91	15.71	**0.00**+	14.30	**0.00**+	15.96	**0.00**+	7.30	0.41
	5	6.82	6.82	1.00	11.84	**0.00**+	9.79	**0.00**+	6.71	0.86	13.62	**0.00**+	13.51	**0.00**+	13.96	**0.00**+	6.65	0.78
	6	6.68	6.70	0.98	6.68	1.00	6.44	0.67	6.63	0.92	11.16	**0.00**+	8.62	**0.04**+	11.17	**0.00**+	6.50	0.63
	7	6.20	6.20	1.00	6.20	1.00	6.20	1.00	6.20	1.00	6.20	1.00	6.20	1.00	6.20	1.00	6.20	1.00
	1	6.73	6.67	0.63	6.10	**0.00**−	6.15	**0.00**−	6.27	**0.00**−	7.15	**0.00**+	6.97	**0.01**+	7.12	**0.00**+	6.18	**0.00**−
	2	4.95	4.95	0.95	6.02	**0.00**+	5.90	**0.00**+	4.97	0.51	6.37	**0.00**+	6.82	**0.00**+	6.67	**0.00**+	4.95	0.92
	3	4.74	4.74	1.00	5.39	**0.00**+	5.62	**0.00**+	4.80	0.05	5.16	**0.00**+	6.29	**0.00**+	5.96	**0.00**+	4.87	**0.00**+
abalone	4	4.69	4.69	0.99	5.39	**0.00**+	5.41	**0.00**+	4.72	0.42	4,83	**0.00**+	5.87	**0.00**+	5.73	**0.00**+	4.79	**0.00**+
	5	4.67	4.67	0.95	5.34	**0.00**+	5.29	**0.00**+	4.66	0.88	4.73	0.17	5.29	**0.00**+	5.28	**0.00**+	4.76	**0.01**+
	6	4.64	4.64	0.87	5.21	**0.00**+	5.28	**0.00**+	4.63	0.67	4.71	0.16	4.89	**0.00**+	4.88	**0.00**+	4.70	0.06
	7	4.62	4.62	0.98	4.59	0.32	4.90	**0.00**+	4.60	0.62	4.63	0.78	4.71	0.07	4.63	0.79	4.67	0.12
	8	4.57	4.57	1.00	4.58	1.00	4.57	1.00	4.57	1.00	4.58	1.00	4.58	1.00	4.58	1.00	4.58	1.00
	2	40.39	64.81	**0.00**+	297.51	**0.00**+	293.6	**0.00**+	75.45	**0.00**+	276.56	**0.00**+	141.00	**0.00**+	295.11	**0.00**+	291.26	**0.00**+
	4	18.99	19.33	0.55	279.65	**0.00**+	82.44	**0.00**+	60.09	**0.00**+	242.23	**0.00**+	32.66	0.15	222.18	**0.00**+	247.39	**0.00**+
	6	19.20	19.22	0.97	116.14	**0.00**+	28.57	0.32	39.89	**0.00**+	167.24	**0.00**+	16.60	0.05	112.87	**0.00**+	206.61	**0.00**+
cpusmall	8	20.66	21.28	0.32	19.69	0.07	20.49	0.78	29.36	**0.00**+	19.96	0.25	17.54	0.06	78.51	**0.00**+	124.44	**0.00**+
	10	21.64	22.52	0.24	20.68	0.15	22.49	0.28	25.61	**0.00**+	20.81	0.25	19.67	0.07	55.55	**0.00**+	59.30	**0.00**+
	12	23.78	23.78	1.00	23.78	1.00	23.78	1.00	23.78	1.00	23.78	1.00	23.78	1.00	23.78	1.00	23.78	1.00
	2	19.00	19.00	1.00	29.36	**0.00**+	19.00	1.00	28.99	**0.00**+	64.09	**0.00**+	62.60	**0.00**+	46.80	**0.00**+	19.00	1.00
	4	16.00	15.94	0.98	25.46	**0.00**+	14.86	0.60	15.19	0.71	38.98	**0.00**+	56.52	**0.00**+	23.22	**0.01**+	13.97	0.35
	6	13.74	13.59	0.94	16.28	0.26	13.90	0.94	13.69	0.98	28.96	**0.00**+	50.93	**0.00**+	18.33	**0.03**+	12.63	0.54
housing	8	11.47	12.46	0.54	15.24	0.06	11.54	0.96	12.02	0.74	24.63	**0.00**+	43.99	**0.00**+	11.38	0.95	11.34	0.93
	10	9.57	10.76	0.40	11.32	0.18	10.49	0.50	11.08	0.28	12.25	0.07	37.94	**0.00**+	11.71	0.15	11.60	0.18
	12	10.12	10.12	1.00	9.45	0.62	9.51	0.65	10.36	0.87	10.81	0.63	17.83	**0.00**+	10.81	0.65	10.69	0.70
	13	10.48	10.48	1.00	10.48	1.00	10.48	1.00	10.48	1.00	10.48	1.00	10.48	1.00	10.48	1.00	10.48	1.00
	2	0.00022	0.00022	0.91	0.00017	**0.00**−	0.00022	0.91	0.00022	0.91	0.00021	0.51	0.00026	0.08	0.00032	**0.00**+	0.00018	**0.00**−
	4	0.00018	0.00018	0.93	0.00016	0.07	0.00025	**0.00**+	0.00017	0.19	0.00021	0.11	0.00023	**0.02**+	0.00020	0.28	0.00022	**0.04**+
	6	0.00021	0.00021	1.00	0.00019	0.08	0.00026	**0.00**+	0.00020	0.29	0.00021	0.88	0.00021	0.16	0.00019	0.12	0.00024	0.06
bodyfat	8	0.00020	0.00020	0.97	0.00023	0.04	0.00026	0.05	0.00020	0.95	0.00022	0.31	0.00023	0.09	0.00019	0.54	0.00025	**0.00**+
	10	0.00020	0.00020	0.99	0.00023	0.05	0.00025	0.05	0.00020	0.95	0.00022	0.14	0.00023	0.12	0.00019	0.78	0.00024	**0.01**+
	12	0.00021	0.00021	1.00	0.00023	0.16	0.00025	0.05	0.00020	0.66	0.00023	0.27	0.00022	0.48	0.00020	0.59	0.00023	0.19
	14	0.00021	0.00021	1.00	0.00021	1.00	0.00021	1.00	0.00021	1.00	0.00021	1.00	0.00021	1.00	0.00021	1.00	0.00021	1.00
	1	0.020	0.020	1.00	0.020	0.95	0.021	0.95	0.021	0.69	0.021	0.65	0.021	0.65	0.021	0.65	0.021	0.85
	10	0.018	0.017	0.92	0.017	0.84	0.019	0.63	0.018	0.80	0.020	0.25	0.021	0.18	0.020	0.20	0.018	0.89
	20	0.017	0.017	0.98	0.018	0.75	0.017	0.89	0.017	0.87	0.020	0.15	0.021	0.11	0.020	0.14	0.017	0.93
triazines	30	0.017	0.018	0.83	0.018	0.63	0.017	0.94	0.017	0.95	0.019	0.30	0.020	0.17	0.020	0.23	0.018	0.97
	40	0.018	0.018	0.94	0.018	0.98	0.018	0.75	0.017	0.85	0.018	0.83	0.019	0.43	0.019	0.46	0.018	0.94
	50	0.018	0.018	0.99	0.018	0.91	0.020	0.52	0.018	0.93	0.018	0.93	0.019	0.73	0.019	0.72	0.018	0.96
	60	0.018	0.018	1.00	0.018	1.00	0.018	1.00	0.018	1.00	0.018	1.00	0.018	1.00	0.018	1.00	0.018	1.00

(Continued.)

TABLE III *(Continued.)*

Dataset	N	SD-L-RFE	SD-G-RFE		mRMR		HSIC-RFE		$\Delta\|\omega\|^2$-RFE		RMB-RFE		SpanB-RFE		RMB-RFE*		SpanB-RFE*	
		Mean value	Mean value	$p-$ value	Mean value	$p-$ value	Mean value	$p-$ value	Mean value	$p-$ value	Mean value	$p-$ value	Mean value	$p-$ value	Mean value	$p-$ value	Mean value	$p-$ value
SCC measure																		
mpg	1	0.73	0.73	1.00	0.72	0.75	0.63	**0.00**+	0.73	1.00	0.63	**0.00**+	0.48	**0.00**+	0.63	**0.00**+	0.72	0.69
	2	0.87	0.87	1.00	0.73	**0.00**+	0.70	**0.00**+	0.87	1.00	0.70	**0.00**+	0.69	**0.00**+	0.71	**0.00**+	0.86	0.25
	3	0.89	0.89	1.00	0.74	**0.00**+	0.74	**0.00**+	0.88	0.22	0.71	**0.00**+	0.73	**0.00**+	0.71	**0.00**+	0.87	0.15
	4	0.89	0.89	1.00	0.78	**0.00**+	0.78	**0.00**+	0.89	0.91	0.74	**0.00**+	0.76	**0.00**+	0.74	**0.00**+	0.88	0.41
	5	0.89	0.89	1.00	0.81	**0.00**+	0.84	**0.00**+	0.89	0.86	0.78	**0.00**+	0.78	**0.00**+	0.86	**0.00**+	0.89	0.78
	6	0.89	0.89	0.98	0.89	1.00	0.90	0.67	0.89	0.92	0.82	**0.00**+	0.86	**0.04**+	0.82	**0.00**+	0.90	0.63
	7	0.90	0.89	1.00	0.90	1.00	0.89	1.00	0.89	1.00	0.90	1.00	0.90	1.00	0.90	1.00	0.90	1.00
abalone	1	0.36	0.36	0.63	0.42	**0.00**−	0.41	**0.00**−	0.40	**0.00**−	0.32	**0.00**+	0.33	**0.01**+	0.32	**0.00**+	0.41	**0.00**−
	2	0.53	0.53	0.95	0.42	**0.00**+	0.44	**0.00**+	0.53	0.51	0.39	**0.00**+	0.35	**0.00**+	0.36	**0.00**+	0.53	0.92
	3	0.55	0.55	1.00	0.49	**0.00**+	0.46	**0.00**+	0.54	0.05	0.51	**0.00**+	0.40	**0.00**+	0.43	**0.00**+	0.54	**0.00**+
	4	0.55	0.55	0.99	0.49	**0.00**+	0.48	**0.00**+	0.55	0.42	0.54	**0.02**+	0.44	**0.00**+	0.45	**0.00**+	0.54	**0.00**+
	5	0.55	0.56	0.95	0.49	**0.00**+	0.50	**0.00**+	0.56	0.88	0.55	0.17	0.50	**0.00**+	0.50	**0.00**+	0.55	**0.01**+
	6	0.56	0.56	0.87	0.50	**0.00**+	0.50	**0.00**+	0.56	0.67	0.55	0.16	0.53	**0.00**+	0.54	**0.00**+	0.55	0.06
	7	0.56	0.56	0.98	0.56	0.32	0.53	**0.00**+	0.56	0.62	0.56	0.78	0.55	0.07	0.56	0.78	0.53	0.12
	8	0.56	0.56	1.00	0.56	1.00	0.56	1.00	0.56	1.00	0.56	1.00	0.56	1.00	0.56	1.00	0.56	1.00
cpusmall	2	0.89	0.82	**0.00**+	0.16	**0.00**+	0.17	**0.00**+	0.79	**0.00**+	0.22	**0.00**+	0.60	**0.00**+	0.17	**0.00**+	0.17	**0.00**+
	4	0.95	0.95	0.55	0.21	**0.00**+	0.77	**0.00**+	0.83	**0.00**+	0.31	**0.00**+	0.91	0.15	0.37	**0.00**+	0.29	**0.00**+
	6	0.95	0.95	0.97	0.67	**0.00**+	0.92	0.32	0.89	**0.00**+	0.52	**0.00**+	0.95	0.05	0.68	**0.00**+	0.41	**0.00**+
	8	0.94	0.94	0.32	0.94	0.07	0.94	0.78	0.92	**0.00**+	0.94	0.25	0.95	0.06	0.78	**0.00**+	0.65	**0.00**+
	10	0.94	0.94	0.24	0.94	0.15	0.94	0.28	0.93	**0.00**+	0.94	0.25	0.94	0.07	0.84	**0.00**+	0.84	**0.00**+
	12	0.93	0.93	1.00	0.93	1.00	0.93	1.00	0.93	1.00	0.93	1.00	0.93	1.00	0.93	1.00	0.93	1.00
housing	2	0.77	0.77	1.00	0.65	**0.00**+	0.77	1.00	0.65	**0.00**+	0.23	**0.00**+	0.25	**0.00**+	0.45	**0.00**+	0.77	1.00
	4	0.80	0.80	0.98	0.70	**0.00**+	0.82	0.60	0.81	0.71	0.54	**0.00**+	0.34	**0.00**+	0.73	**0.01**+	0.83	0.35
	6	0.83	0.83	0.94	0.80	0.26	0.83	0.94	0.83	0.98	0.66	**0.00**+	0.41	**0.00**+	0.79	**0.03**+	0.84	0.54
	8	0.86	0.85	0.54	0.82	0.06	0.86	0.96	0.85	0.74	0.71	**0.00**+	0.49	**0.00**+	0.86	0.95	0.86	0.93
	10	0.88	0.87	0.40	0.86	0.18	0.87	0.50	0.86	0.28	0.85	0.07	0.56	**0.00**+	0.86	0.15	0.86	0.18
	12	0.88	0.88	1.00	0.88	0.62	0.88	0.65	0.87	0.87	0.86	0.63	0.79	**0.00**+	0.87	0.64	0.86	0.70
	13	0.87	0.87	1.00	0.87	1.00	0.87	1.00	0.87	1.00	0.87	1.00	0.87	1.00	0.87	1.00	0.87	1.00
bodyfat	2	0.89	0.89	0.91	0.95	**0.00**−	0.89	0.91	0.89	0.91	0.52	0.51	0.38	0.08	0.18	**0.00**+	0.79	**0.00**+
	4	0.84	0.84	0.93	0.92	0.07	0.83	**0.00**+	0.86	0.19	0.73	0.11	0.46	**0.02**+	0.58	0.28	0.75	**0.04**+
	6	0.79	0.79	1.00	0.84	0.08	0.80	**0.00**+	0.81	0.29	0.79	0.88	0.47	0.16	0.80	0.12	0.75	0.06
	8	0.80	0.80	0.97	0.79	0.05	0.79	0.05	0.78	0.95	0.79	0.31	0.48	0.09	0.78	0.54	0.73	**0.00**+
	10	0.75	0.75	0.99	0.76	0.05	0.77	0.05	0.76	0.95	0.78	0.14	0.53	0.12	0.76	0.78	0.73	**0.01**+
	12	0.74	0.74	1.00	0.73	0.16	0.76	0.05	0.75	0.66	0.76	0.27	0.57	0.48	0.75	0.59	0.75	0.19
	14	0.73	0.73	1.00	0.73	1.00	0.73	1.00	0.73	1.00	0.73	1.00	0.73	1.00	0.73	1.00	0.73	1.00
triazines	1	0.12	0.12	1.00	0.08	0.95	0.08	0.95	0.07	0.69	0.094	0.65	0.11	0.85	0.094	0.65	0.11	0.85
	10	0.26	0.27	0.92	0.25	0.84	0.19	0.63	0.22	0.80	0.11	0.25	0.11	0.18	0.12	0.20	0.26	0.89
	20	0.28	0.29	0.98	0.22	0.75	0.26	0.89	0.28	0.87	0.12	0.15	0.12	0.11	0.14	0.14	0.30	0.93
	30	0.29	0.26	0.83	0.20	0.62	0.26	0.94	0.29	0.95	0.18	0.30	0.14	0.17	0.17	0.23	0.28	0.97
	40	0.26	0.26	0.94	0.26	0.98	0.22	0.75	0.27	0.85	0.25	0.83	0.17	0.43	0.17	0.46	0.27	0.94
	50	0.25	0.25	0.99	0.26	0.90	0.17	0.52	0.26	0.93	0.22	0.94	0.19	0.73	0.21	0.72	0.26	0.96
	60	0.25	0.25	1.00	0.25	1.00	0.25	1.00	0.25	1.00	0.25	1.00	0.25	1.00	0.25	1.00	0.25	1.00

advantages of the proposed methods are obvious. Specifically, the proposed methods perform better than RMB-RFE* and Span-RFE* when $|\mathcal{D}_{trn}| = 100, 70$, better than HSIC-RFE and $\Delta\|\omega\|^2$-RFE when $|\mathcal{D}_{trn}| = 50, 40$, and better than mRMR for all $|\mathcal{D}_{trn}|$. This can be verified by aforementioned t-test. Also, it is interesting to see that the curves yielded by SD-L-RFE and SD-G-RFE constantly have one minimal MSE point (or maximal for SCC), and the unique extreme point happens when the top two features are selected. These bimodal curves strongly validate the effectiveness of the proposed feature selection methods. This is not the case for the other methods. The figures for other two problems show similar patterns and therefore not shown here.

B. Real Problems

Six real-world datasets from the Statlib,[1] UCI repository [25], and Delve archive[2] are used for evaluation purposes. Description of these datasets and the parameters used in the experiments are given in Table II.

Table III shows the t-test results for all six real-world datasets. It is seen from this table that the proposed methods consistently perform at least as well as, if not better than, all benchmark methods and the advantage is more significant for mpg, abalone, cpusmall, Housing, and Bodyfat datasets. There are two exceptions: the first few rows of the datasets abalone and bodyfat show that the SD-L-RFE is statistically worse off than some benchmark methods. This should not be seen as a worrying sign, as it happens for the case where one or two features are used. Clearly, this case corresponds to one of overelimination of features. In practice, early stopping of RFE would have been triggered by the substantial increase of MSE or decrease of SCC.

C. Discussion

The better performance of the proposed method over mRMR is expected, since this common filter method is not effective in capturing effects of three or more interacting features. The other filter method, i.e., HSIC-RFE, appears to be quite effective in dealing with data having interacting features, and generally shows nearly comparable performance with the wrapper method $\Delta\|\omega\|^2$-RFE. However, it is not as effective as the proposed methods from the results on artificial problems, especially when the training data is sparse, and on real-world datasets of mpg, abalone and cputime. The better performance of the proposed methods over $\Delta\|\omega\|^2$-RFE, RMB-RFE and Span-RFE is interesting and deserves more attention, since all of them are wrapper-based feature selection methods for SVR. The better performance of the proposed methods over them is probably due to the following two differences: First, different ranking criteria are used. The proposed method uses the "aggregate" sensitivity of SVR probabilistic predictions with respect to a feature over the feature space, while $\Delta\|\omega\|^2$-RFE uses the sensitivity of the cost function of SVR with respect to a feature and RMB-RFE and Span-RFE uses the sensitivity of the error bound of SVR with respect to a feature. Second, $\Delta\|\omega\|^2$-RFE, RMB-RFE, and Span-RFE assume that the SVR solution remains unchanged when a feature is removed within each RFE iteration. This appears to be a strong assumption, judging from the relative performances of RMB-RFE, Span-RFE RMB-RFE*, and Span-RFE*.

Another advantage of the proposed method is the modest computational load. As mentioned in Section III, the evaluation of scores for d features includes a one-time training of SVR of about $O(n^{2.3})$ [26] complexity, one-time evaluation of σ^L (or σ^G) of $O(mn)$ where $n = |\mathcal{D}|$, m is the number of support vectors, d-time RP process of $O(dn)$, d-time evaluation of $\sigma^L_{(j)}$ (or $\sigma^G_{(j)}$) of $O(dmn)$, and d-time evaluation of D_{KL} of $O(dn)$. Hence, after one-time training of SVR, the proposed criterion scales linearly with respect to d and n. Obviously, $\Delta\|\omega\|^2$-RFE, RMB-RFE, and Span-RFE have similar computational costs as the proposed methods. However, RMB-RFE* and Span-RFE* require the training of SVR $d-1$ times more than the proposed methods when evaluating the scores for the d features. This additional computational load is of $O(dn^{2.3})$, which is significant when n is large.

VI. Conclusion

This brief presented a new wrapper-based feature selection method for SVR. This method measures the importance of a feature by the aggregation, over the feature space, of the sensitivity of SVR probabilistic prediction with and without the feature. Two approximations of the criterion with RP process were proposed. The numerical experiments on both artificial and real-world problems suggest that the proposed method generally performs as least as well as, if not better than, three benchmark methods. The advantage of the proposed methods is more significant when the training data is sparse or has a low samples-to-features ratio. As a wrapper method, the computational cost of proposed methods is moderate.

[1] Available at http://lib.stat.cmu.edu/datasets/.

[2] Available at http://www.cs.toronto.edu/~delve/data/datasets.html.

References

[1] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artif. Intell.*, vol. 97, nos. 1–2, pp. 245–271, Dec. 1997.

[2] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, nos. 1–2, pp. 273–324, Dec. 1997.

[3] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, nos. 1–3, pp. 389–422, 2002.

[4] L. Song, A. Smola, A. Gretton, J. Bedo, and K. Borgwardt, "Supervised feature selection via dependence estimation," in *Proc. 24th Int. Conf. Mach. Learn.*, Corvallis, OR, 2007, pp. 823–830.

[5] A. Rakotomamonjy, "Analysis of SVM regression bounds for variable ranking," *Neurocomputing*, vol. 70, nos. 7–9, pp. 1489–1501, Mar. 2007.

[6] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal Stat. Soc. Ser. B*, vol. 58, no. 1, pp. 267–288, 1996.

[7] A. Rakotomamonjy, "Variable selection using SVM based criteria," *J. Mach. Learn. Res.*, vol. 3, pp. 1357–1320, Mar. 2003.

[8] K.-Q. Shen, C.-J. Ong, X.-P. Li, and E. P. Wilder-Smith, "Feature selection via sensitivity analysis of SVM probabilistic outputs," *Mach. Learn.*, vol. 70, no. 1, pp. 1–20, 2008.

[9] J.-B. Yang, K.-Q. Shen, C.-J. Ong, and X.-P. Li, "Feature selection for MLP neural network: The use of random permutation of probabilistic outputs," *IEEE Trans. Neural Netw.*, vol. 20, no. 12, pp. 1911–1922, Dec. 2009.

[10] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Stat. Comput.*, vol. 14, no. 3, pp. 199–222, 2004.

[11] D. J. C. MacKay, "The evidence framework applied to classification networks," *Neural Comput.*, vol. 4, no. 5, pp. 720–736, Sep. 1992.
[12] M. H. Law and J. T. Kwok, "Bayesian support vector regression," in *Proc. 8th Int. Workshop Artif. Intell. Stat.*, 2001, pp. 239–244.
[13] J. B. Gao, S. R. Gunn, C. J. Harris, and M. Brown, "A probabilistic framework for SVM regression and error bar estimation," *Mach. Learn.*, vol. 46, nos. 1–3, pp. 71–89, 2002.
[14] W. Chu, S. S. Keerthi, and C. J. Ong, "Bayesian support vector regression using a unified loss function," *IEEE Trans. Neural Netw.*, vol. 15, no. 1, pp. 29–44, Jan. 2004.
[15] C. J. Lin and R. C. Weng, "Simple probabilistic predictions for support vector regression," Dept. Comput. Sci., Nat. Taiwan Univ., Taipei, Taiwan, Tech. Rep., 2004.
[16] C. M. Bishop, *Neural Networks for Pattern Recognition*. London, U.K.: Oxford Univ. Press, Nov. 1995.
[17] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Trans. Neural Netw.*, vol. 5, no. 4, pp. 537–550, Jul. 1994.
[18] F. H. Long, H. C. Peng, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.
[19] E. S. Page, "A note on generating random permutations," *Appl. Stat.*, vol. 16, no. 3, pp. 273–274, 1967.
[20] W. D. Penny, "KL divergences of normal, gamma, Dirichlet and Wishart densities," Dept. Cognit. Neurol., Univ. College London, London, U.K., Tech. Rep., Mar. 2001.
[21] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with Hilbert–Schmidt norms," in *Proc. 16th Int. Conf. Algorith. Learn. Theory*, Oct. 2005, pp. 63–78.
[22] A. P. A. Silva, V. H. Ferreira, and R. M. Velasquez, "Input space to neural network based load forecasters," *Int. J. Forecast.*, vol. 24, no. 4, pp. 616–629, Oct.–Dec. 2008.
[23] C.-C. Chang and C.-J. Lin. (2001). *LIBSVM: A Library for Support Vector Machines* [Online]. Available: http://www.csie.ntu.edu.tw/∼cjlin/libsvm
[24] J. H. Friedman, "Multivariate adaptive regression splines," *Ann. Stat.*, vol. 19, no. 1, pp. 1–67, 1991.
[25] A. Asuncion and D. J. Newman. (2007). *UCI Machine Learning Repository* [Online]. Available: http://www.ics.uci.edu/∼mlearn/MLRepository.html
[26] J. C. Platt, "Using sparseness and analytic QP to speed training of support vector machines," in *Advances in Neural Information Processing Systems 11*, M. S. Kearns, S. A. Solla, and D. A. Cohn, Eds. Cambridge, MA: MIT Press, 1998.

Improvements on Twin Support Vector Machines

Yuan-Hai Shao, Chun-Hua Zhang, Xiao-Bo Wang, and Nai-Yang Deng

***Abstract*— For classification problems, the generalized eigenvalue proximal support vector machine (GEPSVM) and twin support vector machine (TWSVM) are regarded as milestones in the development of the powerful SVMs, as they use the nonparallel hyperplane classifiers. In this brief, we propose an improved version, named twin bounded support vector machines (TBSVM), based on TWSVM. The significant advantage of our TBSVM over TWSVM is that the structural risk minimization principle is implemented by introducing the regularization term. This embodies the marrow of statistical learning theory, so this modification can improve the performance of classification. In addition, the successive overrelaxation technique is used to solve the optimization problems to speed up the training procedure. Experimental results show the effectiveness of our method in both computation time and classification accuracy, and therefore confirm the above conclusion further.**

***Index Terms*— Machine learning, maximum margin, structural risk minimization principle, support vector machines.**

Manuscript received September 13, 2009; revised February 11, 2011; accepted March 6, 2011. Date of publication May 5, 2011; date of current version June 2, 2011. This work is supported in part by the National Natural Science Foundation of China, under Grant 10971223 and Grant 11071252.

Y.-H. Shao and N.-Y. Deng are with the College of Science China Agricultural University, Beijing 100083, China (e-mail: shaoyuanhai21@163.com; dengnaiyang@cau.edu.cn).

C.-H. Zhang is with the Department of Mathematics, Information School, Renmin University of China, Beijing 100872, China (zhangchunhua@ruc.edu.cn).

X.-B. Wang is with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: xb-wang10@mails.tsinghua.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2011.2130540

I. Introduction

Support vector machines (SVMs), being computationally powerful tools for supervised learning [1]–[3], have already outperformed most other systems in a wide variety of applications [4]–[6]. For the standard support vector classification (SVC), its primal problem can be understood in the following way: construct two parallel support hyperplanes such that, on one hand, the band between the two parallel hyperplanes separates the two classes (the positive and negative data points) well, on the other hand, the width between the two hyperplanes is maximized, leading to the introduction of a regularization term. Thus, the structural risk minimization principle is implemented. The final separating hyperlane is selected to be the "middle one" between the two hyperplanes. Different from SVC with two parallel hyperplanes, some nonparallel hyperplane classifiers such as the generalized eigenvalue proximal support vector machine (GEPSVM) and twin support vector machine (TWSVM) have been proposed in [7] and [8]. TWSVM seeks two nonparallel proximal hyperplanes such that each hyperplane is closest to one of two classes and as far as possible from the other class. A fundamental difference between TWSVM and SVC is that TWSVM solves two smaller sized quadratic programming problems (QPPs), whereas SVC solves one larger QPP. Therefore, TWSVM works faster than SVC. Experimental results in [8], and [9] have shown the effectiveness of TWSVM over both standard SVC and GEPSVM on UCI datasets. In addition, TWSVM is excellent at dealing with the "Cross Planes" dataset. Thus, the methods of constructing the nonparallel hyperplanes have been studied extensively [9]–[12].

It is well known that one significant advantage of SVC is the implementation of the structural risk minimization principle [13], [14]. However, only the empirical risk is considered in the primal problems of TWSVM. In addition, we noticed that the inverse matrices $(G^\top G)^{-1}$ and $(H^\top H)^{-1}$ appear in the dual problems. This implies that, in order to obtain the dual problems, TWSVM must assume that the inverse matrices $(G^\top G)^{-1}$ and $(H^\top H)^{-1}$ exist or the matrices $G^\top G$ and $H^\top H$ are nonsingular. However, this extra prerequisite cannot always be satisfied. So the duality theory in TWSVM is not perfect from the theoretical point of view, although these inverse matrices have been handled by modifying the dual problems technically and elegantly.