Feature Selection for Support Vector Regression Using Probabilistic Prediction

Jian-Bo Yang
Department of Mechanical Engineering
National University of Singapore
Singapore 117576
yangjianbo@nus.edu.sg

Chong-Jin Ong
Department of Mechanical Engineering
National University of Singapore
Singapore 117576
mpeongcj@nus.edu.sg.

ABSTRACT

This paper presents a novel 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 experiment shows that the proposed method generally performs better, and at least as well as the existing methods, with notable advantage when the data set is sparse.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications-data mining; I.2.6 [**Artificial Intelligence**]: Learning- induction

General Terms

Algorithms

Keywords

Support vector regression, feature selection, feature ranking, probabilistic predictions, random permutation.

1. INTRODUCTION

Feature selection plays an important role in pattern recognition, data mining, information retrieval and has been the subject of intense research in the past decade, see, for example, [9, 10] and the references therein. Generally, methods for feature selection can be classified into two categories: filter and wrapper methods [10]. Wrapper methods rely heavily on the specific structure of the underlying learning algorithm while 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 [11, 22, 17, 23], 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 [11, 20], 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 paper proposes a new wrapper-based feature selection method for SVR, motivated by our earlier work on classification problem using Support Vector Machine (SVM) [21] and Multi-Layer Perceptrons (MLP) neural networks [25]. 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 data set. 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 computations of proposed method is expensive, two approximations are proposed. Each of the two approximations, embedded in an overall feature selection scheme, is tested on artificial and real-world data sets and compared with several other existing feature selection methods. The experimental result shows that the proposed method performs generally better, if not at least as well, than other methods in almost all experiments.

The following notations are used: data set $\mathcal{D} = \{x_i, y_i\}, i \in \mathcal{I}_\mathcal{D}$ is assumed given with $x_i \in R^d$ being the i^{th} sample having d features, $y_i \in R$ is the corresponding output and $\mathcal{I}_\mathcal{D}$ is the set of indices of samples of $\mathcal{D}$ with $|\mathcal{I}_\mathcal{D}|$ being its cardinality; for convenience, $|\mathcal{D}| = |\mathcal{I}_\mathcal{D}|$; $x_i^j \in R$ is the value of the j^{th} feature of the i^{th} sample; the double subscripted symbol $x_{-j,i} \in R^{d-1}$ refers to the i^{th} sample after the j^{th} feature has been removed from x_i. Equivalently, $x_{-j,i} = Z_j^d x_i$ where Z_j^d is the $(d-1) \times d$ matrix obtained by removing the j^{th} row of the $d \times d$ identity matrix. When v is a vector of appropriate dimension, v' is its transpose.

This paper is organized as follow: Section 2 reviews the formulation of probabilistic SVR. Details of the proposed feature ranking criterion and the two approximations are

KDD'10, July 25–28, 2010, Washington, DC, USA.

presented in Section 3. Section 4 shows the overall feature selection scheme. Result of numerical experiment of the proposed method, benchmark against other methods, are reported in Section 5. Section 6 concludes the paper.

2. REVIEW OF PROBABILISTIC SVR

Standard SVR [24] obtains the regressor function, $f(x) := \omega'\phi(x) + b$, for a data set $\mathcal{D}$ by solving the following Primal Problem (PP) over ω, b, ξ, ξ^*:

$$\min \quad \frac{1}{2}\omega'\omega + C\sum_{i\in\mathcal{I_D}}(\xi_i + \xi_i^*) \tag{1}$$

$$s.t. \quad y_i - \omega'\phi(x_i) - b \le \epsilon + \xi_i, \quad \forall i \in \mathcal{I_D} \tag{2}$$

$$\omega'\phi(x_i) + b - y_i \le \epsilon + \xi_i^*, \quad \forall i \in \mathcal{I_D} \tag{3}$$

$$\xi_i, \xi_i^* \ge 0, \qquad \forall i \in \mathcal{I_D} \tag{4}$$

where x is mapped into a high dimensional Hilbert space, $\mathcal{H}$, by the function $\phi : R^d \to \mathcal{H}$. Here, $\omega \in \mathcal{H}$, $b \in R$ are variables that define $f(x)$ and ξ_i, ξ_i^* are the non-negative slack variables needed for enforcing constraints (2) and (3). The regularization parameter, $C > 0$, tradeoffs the size of ω and the amount of slacks while 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):

$$\begin{aligned} \max_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} \quad & -\frac{1}{2}\sum_{i\in\mathcal{I_D}}\sum_{j\in\mathcal{I_D}}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(x_i, x_j) - \\ & \epsilon\sum_{i\in\mathcal{I_D}}(\alpha_i + \alpha_i^*) + \sum_{i\in\mathcal{I_D}} y_i(\alpha_i - \alpha_i^*) \\ s.t. \quad & \sum_{i\in\mathcal{I_D}}(\alpha_i - \alpha_i^*) = 0, \\ & 0 \le \alpha_i \le C, \quad 0 \le \alpha_i^* \le C, \quad i \in \mathcal{I_D} \end{aligned} \tag{5}$$

where α_i and α_i^* are the respective Lagrange multipliers of (2) and (3),

$$\omega = \sum_{i\in\mathcal{I_D}}(\alpha_i - \alpha_i^*)\phi(x_i) \tag{6}$$

and $K(x_i, x_j) = \phi(x_i)'\phi(x_j)$. Using these expressions, the regressor function is known to be

$$f(x) = \omega'\phi(x_i) + b = \sum_{i\in\mathcal{I_D}}(\alpha_i - \alpha_i^*)K(x_i, x) + b. \tag{7}$$

Expression (7) 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 [16], Law and Kwok [14] proposed a Bayesian support vector regression (BSVR) formulation incorporating probabilistic information. Gao et al. [7] improved upon BSVR by deriving the evidence and error bar approximation. Chu et al.[4] proposed the use of a unified loss function over the standard ϵ-insensitive loss function and provided better accuracy in evidence evaluation and inferences.

Another approach to obtaining probabilistic output of the regressor is that used in the Neural Networks framework [2]. It assumes that the output of the regressor is corrupted with noise in the form of

$$y = f(x) + \zeta \tag{8}$$

where ζ belongs to the Gaussian distribution. Lin and Weng [15] also considered the case where ζ belongs to the Laplace distribution. Equivalently, this means that density functions of y for a given x are

$$p^L(y|x;\sigma) = \frac{1}{2\sigma}\exp(-\frac{|y - f(x)|}{\sigma}), \tag{9}$$

$$p^G(y|x;\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(y - f(x))^2}{2\sigma^2}) \tag{10}$$

for the Laplace and Gaussian cases respectively. Like the Neural Network approach, the intention is to obtain estimates of σ of (9) and (10) 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

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

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 $L(\sigma)$ [2, 5]. These expressions are

$$\sigma^L = \frac{\sum_{i\in\mathcal{I_D}}|y_i - f(x_i)|}{|\mathcal{I_D}|}, \tag{11}$$

$$(\sigma^G)^2 = \frac{\sum_{i\in\mathcal{I_D}}(y_i - f(x_i))^2}{|\mathcal{I_D}|} \tag{12}$$

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 (9) and (10) for its computation.

3. THE PROPOSED FEATURE SELECTION CRITERION FOR REGRESSION

The proposed method of feature importance relies on measures of difference between two density functions. Our choice of this measure is the Kullback-Leibler divergence (KL divergence), $D_{KL}(\cdot;\cdot)$. Given two distributions $p(y)$ and $q(y)$,

$$D_{KL}(p(y);q(y)) = \int p(y)\log\frac{p(y)}{q(y)}dy. \tag{13}$$

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 cases where symmetry of the arguments is important or that a reference distribution does not exist, modifications to $D_{KL}(\cdot;\cdot)$ can be easily achieved.

In the case of SVR, the density function $p(y|x)$ at any x is assumed to be (9) or (10) with $f(\cdot)$ being the solution obtained from (7). Given $x \in R^d$, $x_{-j} \in R^{d-1}$ can be obtained by removing the j^{th} feature from x, or, equivalently, $x_{-j} = Z_j^d x$. 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

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

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 j^{th} feature. For convenience, (14) is termed SD measure, short for Sensitivity of Density Functions.

In (14), $p(y|x)$ is either (9) or (10) with $f(\cdot)$ of (7) trained on $\mathcal{D}$. Similarly, $p(y|x_{-j})$ is obtained from $f(\cdot)$ trained on the derived dataset $\mathcal{D}_{-j} := \{(x_{-j,i}, y_i)|x_{-j,i} = Z_j^d x_i, \forall(x_i, y_i) \in \mathcal{D}\}$. Thus, evaluations of $S_D(j)$, $j = 1, \cdots, d$ require the training of SVR d times, each with a different $\mathcal{D}_{-j}$. Clearly, this is a computationally expensive process. Following our work in SVM [21], a random permutation (RP) process is used to approximate $p(y|x_{-j})$ such that the retraining of SVR is avoided. The basic idea of RP process is to randomly permute the values of the j^{th} feature in $\mathcal{D}$ while keeping the values of all other features unchanged. Specifically, let $\{\eta_1, \cdots, \eta_{n-1}\}$ where $n = |\mathcal{D}|$ be a set of uniformly distributed random numbers in the interval $(0, 1)$ and $\lfloor \eta \rfloor$ be the largest integer that is less than η. Then, for each i starting from 1 to $n-1$, compute $\ell = \lfloor n \times \eta_i \rfloor + 1$ and swap the values of x_i^j and x_ℓ^j.

Let $x_{(j)} \in R^d$ be the sample derived from x after the RP process on the j^{th} feature and let $p(y|x_{(j)})$ be the conditional density function of y given $x_{(j)}$. When the data set is sufficiently rich, this RP process of feature j would destroy any correlation of feature j with all other features. With this implicit assumption, the following theorem, the proof of which is given in [21], is known.

Theorem 1.

$$p(y|x_{(j)}) = p(y|x_{-j}) \tag{15}$$

In the case when there are very few data points, the equality in (15) becomes an approximation. Our experiment shows that even with sparse data set, the approach based on (15) is effective.

The utility of Theorem 1 is clear. The density function $p(y|x_{-j})$ of (14) can be replaced by $p(y|x_{(j)})$. Such a replacement brings about significant computational advantage since $p(y|x_{(j)})$ can be evaluated from (9) or (10) using $f(x_{(j)})$ obtained from the SVR training using $\mathcal{D}$. This avoids the expensive d-time retraining of SVR on $\mathcal{D}_{-j}$. Correspondingly, (14) can be equivalently stated as:

$$S_D(j) = \int D_{KL}(p(y|x);p(y|x_{(j)}))p(x)dx. \tag{16}$$

Figure 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 (16) is needed, resulting in

$$\hat{S}_D(j) = \frac{1}{|\mathcal{I}_\mathcal{D}|} \sum_{i \in \mathcal{I}_\mathcal{D}} D_{KL}(p(y_i|x_i);p(y_i|x_{(j),i})). \tag{17}$$

When $p(y|x)$ and $p(y|x_{(j)})$ are Laplace functions or Gaussian functions, explicit expressions of $\hat{S}_D(j)$ exist. From (9), $p(y|x) = p^L(y|x;\sigma^L) = \frac{1}{2\sigma^L}\exp(-\frac{|f(x)-y|}{\sigma^L})$ for the case of Laplace function. The KL divergence of two Laplace distributions can be simply derived as

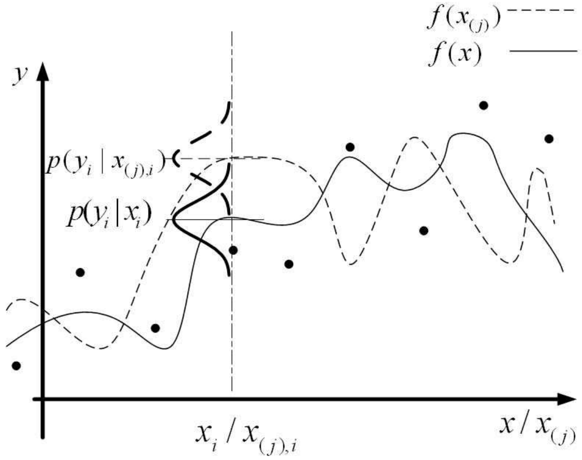

Figure 1: Demonstration of the proposed feature ranking criterion with $d = 1$. Dots indicate locations of y_i

$$\begin{aligned} & D_{KL}(p^L(y|x;\sigma^L);p^L(y|x_{(j)};\sigma^L_{(j)})) \\ & = \ln\frac{\sigma^L_{(j)}}{\sigma^L} - 1 + \frac{\sigma^L}{\sigma^L_{(j)}}\exp(-\frac{|f(x)-f(x_{(j)})|}{\sigma^L}) \\ & + \frac{|f(x)-f(x_{(j)})|}{\sigma^L_{(j)}} \end{aligned} \tag{18}$$

for a given x where σ^L is that given by (11) and $\sigma^L_{(j)}$ is obtained from (11) by replacing $f(x)$ with $f(x_{(j)})$. Using (18) in (17) and removing associated constants yields

$$\begin{aligned} \hat{S}^L_D(j) = \frac{1}{|\mathcal{I}_\mathcal{D}|}\sum_{i\in\mathcal{I}_\mathcal{D}} \Bigg[& \frac{\sigma^L}{\sigma^L_{(j)}}\exp(-\frac{|f(x_i)-f(x_{(j),i})|}{\sigma^L}) + \\ & \frac{|f(x_i)-f(x_{(j),i})|}{\sigma^L_{(j)}} + \ln\frac{\sigma^L_{(j)}}{\sigma^L}\Bigg]. \end{aligned} \tag{19}$$

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

$$\begin{aligned} & D_{KL}(p^G(y|x;\sigma^G);p^G(y|x_{(j)};\sigma^G_{(j)})) \\ & = \frac{f(x)^2 + f(x_{(j)})^2 + (\sigma^G)^2 - 2f(x)f(x_{(j)})}{2(\sigma^G_{(j)})^2} \\ & + \ln\frac{\sigma^G_{(j)}}{\sigma^G} - \frac{1}{2} \end{aligned} \tag{20}$$

and

$$\begin{aligned} \hat{S}^G_D(j) = \frac{1}{2|\mathcal{I}_\mathcal{D}|}\sum_{i\in\mathcal{I}_\mathcal{D}} \Bigg[& \frac{(f(x_i)-f(x_{(j),i}))^2}{(\sigma^G_{(j)})^2} + (\frac{\sigma^G}{\sigma^G_{(j)}})^2 + \\ & 2\ln\frac{\sigma^G_{(j)}}{\sigma^G}\Bigg]. \end{aligned} \tag{21}$$

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

Remark 1. *The kernel matrix is different for each of the d-time evaluation of $\sigma^L_{(j)}$ (or $\sigma^G_{(j)}$) 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 $K(x_{(j),r}, x_{(j),q}) = K(x_r, x_q) + x^j_{(j),r} * x^j_{(j),q} - x^j_r * x^j_q$ for linear kernel and $K(x_{(j),r}, x_{(j),q}) = K(x_r, x_q) * \exp[\kappa(x^j_r - x^j_q)^2 - \kappa(x^j_{(j),r} - x^j_{(j),q})^2]$ with kernel parameter κ for Gaussian kernel.*

Remark 2. *It is possible to develop D_{KL} distance measure that is symmetrical with respect to its arguments. One simple choice is $\frac{1}{2}[D_{KL}(p(y); q(y)) + D_{KL}(q(y); p(y))]$. Following the preceding discussion, the use of such a measure will lead to a equivalently-defined SD metric. Details of such a measure are not included here as numerical experiment shows no significant difference in its performance from that obtained from using the one-sided D_{KL}.*

4. FEATURE SELECTION SCHEME

The proposed $\hat{S}^L_D$ and $\hat{S}^G_D$ 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 removes the least important feature, as determined by $\hat{S}^L_D$ ($\hat{S}^G_D$), recursively from successive trainings of the SVR. Accordingly, the overall scheme with respective to criterion $\hat{S}^L_D$ ($\hat{S}^G_D$) is referred to as the method SD-L-RFE (SD-G-RFE) and its main steps are listed in Algorithm SD-L-RFE. Inputs to Algorithm SD-L-RFE are $\mathcal{D}$ and $\Gamma = \{1, \cdots, d\}$, while the output is a ranked list of features in the form of an index set $\Gamma^\dagger = \{\gamma^\dagger_1, \cdots, \gamma^\dagger_d\}$ where $\gamma^\dagger_j \in \Gamma$ for each $j = 1, \cdots, d$ in decreasing order of importance.

Algorithm SD-L-RFE: Main steps of the feature selection method SD-L-RFE.

```
   Input: $\mathcal{D}$, $\Gamma$
   Output: $\Gamma^\dagger := \{\gamma^\dagger_1, \cdots, \gamma^\dagger_d\}$
 1 while $|\Gamma| > 0$ do
 2     Let $\ell = |\Gamma|$;
 3     if $\ell > 1$ then
 4         Train SVR with $\mathcal{D}$;
 5         For each $j \in \Gamma$, compute $\hat{S}^L_D$ using the criterion
           (19);
 6         Obtained a ranked list $\mathcal{J} = \{j_1, \cdots, j_\ell\}$, $j_k \in \Gamma$
           from $\{\hat{S}^L_D(j)\}^\ell_{j=1}$ such that $\hat{S}^L_D(j_k) \geq \hat{S}^L_D(j_{k+1})$
           for $k = 1, \cdots, \ell - 1$;
 7         Let $\gamma^\dagger_\ell = j_\ell$;
 8         Let $\Gamma = \Gamma \setminus j_\ell$ and $\mathcal{D} = \mathcal{D} \backslash \{x^{j_\ell}_i : i \in \mathcal{I}_\mathcal{D}\}$;
 9     else
10         Let $\gamma^\dagger_1 = j_\ell$;
11     end
12 end
```

As shown above, the while loop at step 1 is invoked d times. Each time except for the last, SVR is trained with a reduced data set $\mathcal{D}$ (step 4) and generates a ranked list $\mathcal{J}$ of all features in $\mathcal{D}$ (step 6) based on the scores of $\hat{S}^L_D$ (step 5). The least important feature (the last element of $\mathcal{J}$) is removed from Γ and stored in the ranked list $\Gamma^\dagger$. The corresponding feature is also removed from the data set $\mathcal{D}$ (step 8). The while loop is then invoked on the reduced sets of Γ and $\mathcal{D}$ again. This process continues, each time removing the least important feature from Γ and storing in the last available position of $\Gamma^\dagger$, until Γ has only one feature, which becomes the most important feature naturally.

Corresponding Algorithm SD-G-RFE involving metric $\hat{S}^G_D$ can be obtained by replacing $\hat{S}^L_D$ by $\hat{S}^G_D$ in steps 5 and 6.

To improve the efficiency of Algorithm SD-L-RFE for large dimensional data set, more than one feature can be removed at one time through a slight modification to step 7 and 8 in the algorithm. Like other wrapper methods, the current algorithm does not involve the re-tuning hyper-parameters of SVR at step 4 for computational consideration.

5. EXPERIMENT

This section presents results of numerical experiment of SD-L-RFE and SD-G-RFE on artificial and real-world data sets. They are also compared against three other existing methods for SVR: the widely-used correlation coefficient method (Corr) [9, 17], a recently proposed dependence maximization method (HSIC) [22] and the well-known SVM-RFE method ($\Delta\|\omega\|^2$) [11, 8]. The first two of them are filter methods while the last is wrapper. All methods, except for Corr since it is not meaningful, are implemented using the RFE scheme described by Algorithm SD-L-RFE for ranking the features. These benchmark methods are hereafter referred to as Corr, HSIC-RFE and $\Delta\|\omega\|^2$-RFE, respectively. There are other regression feature selection methods [23, 18, 13] but they are not included here for comparison because they work for the case of linear kernel.

For each data set, the result of the experiment is reported over 30 realizations, which are created by random (stratified) sampling of the given set $\mathcal{D}$ into subsets $\mathcal{D}_{trn}$ and $\mathcal{D}_{tst}$ for 30 times. As usual, $\mathcal{D}_{trn}$ is used for SVR training, hyper-parameters 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 $K(x_i, x_j) = \exp(-\kappa\|x_i - x_j\|^2)$ where κ is the kernel parameter. In each experiment, all hyper-parameters (C, κ, ϵ) are chosen by a 5-fold cross-validation on the first five realizations of $\mathcal{D}_{trn}$, and the hyper-parameters corresponding to the lowest average cross-validation error among five realizations is chosen. The grid over the (C, κ, ϵ) is $[2^{-2}, 2^{-1}, \cdots, 2^6] \times [2^{-6}, 2^{-5}, \cdots, 2^2] \times [2^{-5}, 2^{-4}, \cdots, 2^2]$.

The well-known regression performance measures, mean squared error (MSE), is used to evaluate the performance. It is given by

$$\text{MSE} := \frac{\sum_{i=1}^{|\mathcal{D}_{tst}|}(\hat{y}_i - y_i)^2}{|\mathcal{D}_{tst}|}, \qquad (22)$$

where y_i and $\hat{y}_i$ are the true and predicted target values respectively .

Using this performance measures, the average MSE among 30 realizations against the number of top-ranked features for each feature selection method is plotted. This is followed by result of statistical paired t-test using MSE for all problems. Specifically, paired t-test between SD-L-RFE and each of the other methods is conducted using different number of top ranked features. Herein, the null hypothesis is that the mean MSE of the two tested methods are 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 [3], where sequential minimal optimization method is used to solve the dual problem (5).

5.1 Artificial Problem

In this subsection, an artificial regression problems is used to evaluate the performance of every feature selection method. This problem is used in [6] and has 10 variables, $x^1, \cdots, x^{10}$, uniformly distributed over the range of [0,1]. The target variable depends only on the first five variables and is given by

$$y = 0.1\exp(4x^1) + \frac{4}{1+\exp(-20(x^2-0.5))} + 3x^3 + 2x^4 + x^5 + \delta \tag{23}$$

where $\delta \sim \mathcal{N}(0, 0.1)$ is a Gaussian random noise. This has 2000 samples and 30 realizations are generated from the 2000 samples by randomly splitting it into $\mathcal{D}_{trn}$ and $\mathcal{D}_{tst}$ with $|\mathcal{D}_{trn}|$:$|\mathcal{D}_{tst}|$=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. More exactly, four settings of decreasing $|\mathcal{D}_{trn}|$ at 200, 100, 70 and 50 are considered in this problem.

Table 1 presents the number of realizations (out of 30 realizations) that feature $1, 2, 3, 4, 5$ are successfully ranked as the first five most important features by the various methods for the four settings of $|\mathcal{D}_{trn}|$. The best performance in each setting is highlighted in bold. From this table, the advantage of the proposed methods over benchmark methods is evident in all settings. Even in the easiest setting of $|\mathcal{D}_{trn}| = 200$, none of benchmark methods are able to produce the correct ranked list for 30 realizations. As the size of $|\mathcal{D}_{trn}|$ decreases, the performance of proposed methods degrades much less than that of benchmark methods.

Table 1: The number of realizations that feature $1, 2, 3, 4, 5$ are successfully ranked in the top five positions over 30 realizations for the artificial problem. The best performance for each $|\mathcal{D}_{trn}|$ is highlighted in bold.

$|\mathcal{D}_{trn}|$ Method	200	100	70	50
Corr	15	8	5	3
HSIC-RFE	14	5	5	3
$\Delta\|\omega\|^2$-RFE	4	5	11	4
SD-L-RFE	**30**	27	21	**19**
SD-G-RFE	**30**	**28**	**23**	**19**

Figure 2 shows the plots of the MSE against the number top-ranked features used in SVR. Again, it shows the advantage of the proposed methods over the benchmark methods. The advantage is more significant when $|\mathcal{D}_{trn}|$ becomes smaller. This performance difference is also statistically significant, as shown in the paired t-tests result of Table 2. Table 2 also shows that SD-L-RFE is significantly better than all benchmark methods for all sizes of $|\mathcal{D}_{trn}|$ while the differences between SD-L-RFE and SD-G-RFE are not significant.

Table 3: Description of real-world data sets. $|\mathcal{D}_{trn}|$, $|\mathcal{D}_{tst}|$, d, C, κ and ϵ refer to the number of training samples, number of test samples, number of features, and SVR hyper-parameters C, κ, ϵ respectively.

Item Dataset	$|\mathcal{D}_{trn}|$	$|\mathcal{D}_{tst}|$	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
pyrim	67	7	27	2^0	2^{-6}	2^{-5}
triazines	168	18	60	2^{-1}	2^{-6}	2^{-3}

5.2 Real Problems

Six real-world data sets from the Statlib[1], UCI repository [1] and Delve archive[2] are used for evaluation purposes. Description of these data sets and the parameters used in the experiments are given in Table 3. Figures 3-8 show MSE against the number of top-ranked features for mpg, abalone, cpusmall, housing, pyrim and triazines respectively. Statistical paired$t$-test is also conducted on all real-world problem. In this paper, we only show the t-test results of problem mpg in Table 4 and abalone in Table 5 under the consideration of limited space.

For the mpg problem, Figure 3 shows the MSE against the number of top-ranked features in SVR for the various methods. It can be observed that given the same number of features used, the proposed methods consistently perform at least as well, if not better than benchmark methods. Specifically, both proposed methods perform significantly better than two filter methods HSIC-RFE and Corr at different number of features, while they perform comparably with $\Delta\|\omega\|^2$-RFE. This is confirmed by the paired t-tests' result in Table 4.

For the other real-world problems (abalone, cpusmall, housing, pyrim and triazines), the experimental result shows similar patterns to that of mpg, as shown in Figures 4 to 8 respectively. Generally, the t-test result shows that the proposed methods almost perform better than the benchmark methods in all real-world problems.

5.3 Discussion

In summary, the effectiveness of the proposed feature selection method is demonstrated for both artificial and real-world problems. In the artificial problem, the proposed method can consistently yield better performance than all three benchmark methods, and the advantage is more evident when $|\mathcal{D}_{trn}|$ is small. This is confirmed by statistical paired t-test results. Furthermore, when the training data become sparse, the performances of proposed methods degrade much less than the benchmark methods. In real-world problems, it can be observed from all plots and t-test results that the proposed methods consistently perform at least as well, if not better than benchmark methods for all problems.

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

[2] http://www.cs.toronto.edu/∼delve/data/datasets.html

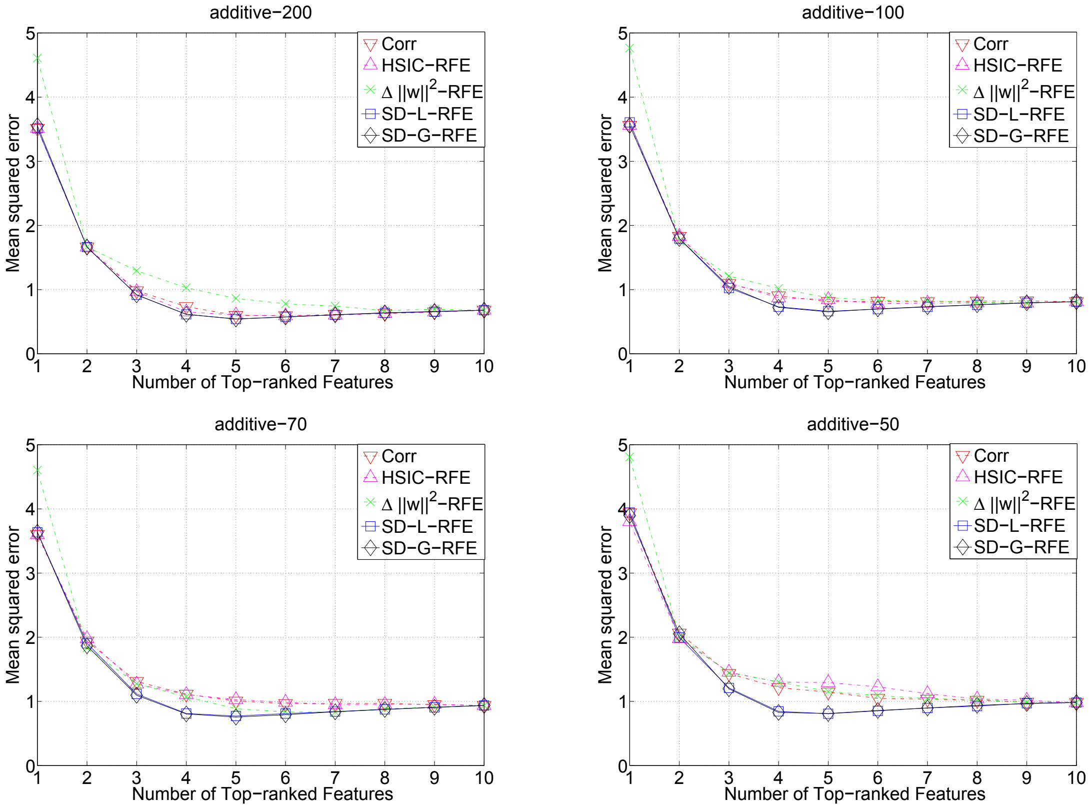

Figure 2: Average MSE against top-ranked features over 30 realizations for the artificial problem with four settings.

Table 2: Result of paired *t*-test using the top 5 features for the artificial problem. *p*-values less than 0.05 are highlighted in bold.

$\lvert\mathcal{D}_{trn}\rvert$	SD-L-RFE	Corr		HSIC-RFE		$\Delta\lVert\omega\rVert^2$-RFE		SD-G-RFE	
	mean value	mean value	p-value	mean value	p-value	mean value	p-value	mean value	p-value
200	0.54	0.60	**0.00+**	0.60	**0.00+**	0.86	**0.00+**	0.54	1.00
100	0.66	0.81	**0.00+**	0.84	**0.00+**	0.88	**0.00+**	0.65	0.59
70	0.77	1.01	**0.00+**	1.03	**0.00+**	0.89	**0.03+**	0.75	0.48
50	0.81	1.15	**0.00+**	1.29	**0.00+**	1.16	**0.00+**	0.81	0.89

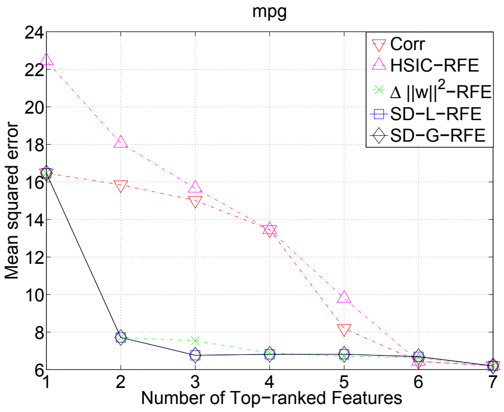

Figure 3: Average MSE against top-ranked features over 30 realizations for mpg

The better performance of proposed method over Corr is expected since this common filter method assumes that all features are independent. The other filter method, 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 the artificial problem, especially when the training data is sparse, and on real-world data sets of mpg, abalone and cputime. The better performance of the proposed methods over $\Delta\|\omega\|^2$-RFE is interesting and deserves more attentions, since both of them are wrapper-based feature selection methods for SVR. These two wrapper methods use the same RFE scheme but differ in their ranking criteria. The proposed method uses the "aggregat" sensitivity of SVR probabilistic predictions with respect to a feature over the feature space as the feature ranking criterion, while $\Delta\|\omega\|^2$-RFE uses the sensitivity of the cost function of SVR with respect to a feature. Another difference is that $\Delta\|\omega\|^2$-RFE has an additional assumption that the SVR solution remains unchanged when one feature is removed while it is unnecessary for the proposed method. These two differences are likely to contribute to the better performance of the proposed methods over $\Delta\|\omega\|^2$-RFE.

Another advantage of the proposed method is the modest computational complexity. As mentioned in Section 3, the evaluation of scores for d features includes a one-time training of SVR about $O(n^2) \sim O(n^3)$ [12, 19], one-time evaluation of σ^L (or σ^G) about $O(mn)$, m is the number of support vectors which is often much less than n, d-time RP process about $O(dn)$, d-time evaluation of $\sigma^L_{(j)}$ (or $\sigma^G_{(j)}$) about $O(dmn)$, and d-time evaluation of D_{KL} about $O(dn)$. Hence, after one-time training SVR, the proposed criterion scales linearly with respect to d and n.

6. CONCLUSIONS

This paper presents 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 random permutation process are proposed. The numerical experiment on both artificial and real-world problems

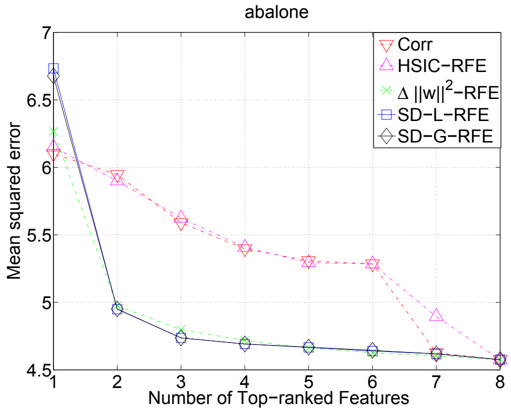

Figure 4: Average MSE against top-ranked features over 30 realizations for abalone

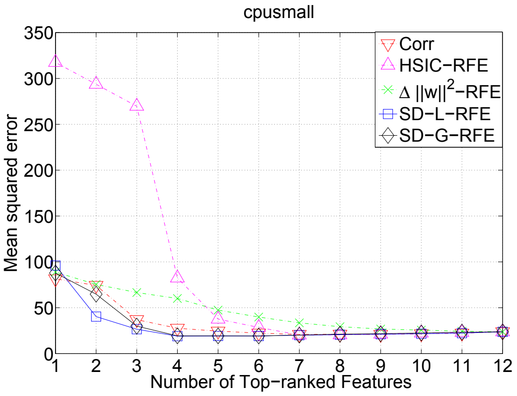

Figure 5: Average MSE against top-ranked features over 30 realizations for cpusmall

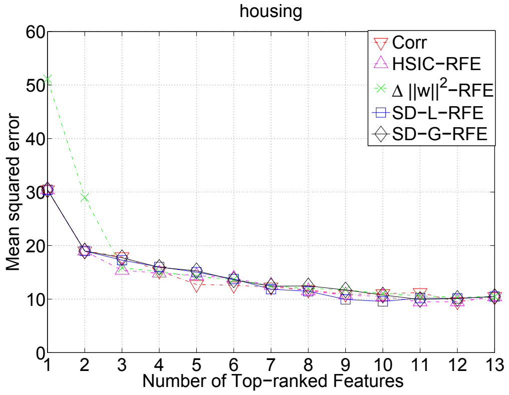

Figure 6: Average MSE against top-ranked features over 30 realizations for housing

Table 4: t-test on mpg data set and the p-values less than 0.05 are highlighted in bold. No. is the number of top ranked features.

No.	SD-L-RFE	Corr		HSIC-RFE		$\Delta\|\omega\|^2$-RFE		SD-G-RFE	
	mean value	mean value	p-value	mean value	p-value	mean value	p-value	mean value	p-value
1	16.47	16.47	1.00	22.45	**0.00+**	16.47	1.00	16.47	1.00
2	7.71	15.85	**0.00+**	18.06	**0.00+**	7.71	1.00	7.71	1.00
3	6.76	15.02	**0.00+**	15.67	**0.00+**	7.54	0.22	6.76	1.00
4	6.81	13.46	**0.00+**	13.46	**0.00+**	6.88	0.91	6.81	1.00
5	6.82	8.18	0.15	9.79	**0.00+**	6.71	0.86	6.82	1.00
6	6.68	6.44	0.67	6.44	0.67	6.63	0.92	6.70	0.98
7	6.20	6.20	1.00	6.20	1.00	6.20	1.00	6.20	1.00

Table 5: t-test on abalone data set and the p-values less than 0.05 are highlighted in bold. No. is the number of top ranked features.

No.	SD-L-RFE	Corr		HSIC-RFE		$\Delta\|\omega\|^2$-RFE		SD-G-RFE	
	mean value	mean value	p-value	mean value	p-value	mean value	p-value	mean value	p-value
1	6.73	6.10	**0.00-**	6.15	**0.00-**	6.27	**0.00-**	6.67	0.63
2	4.95	5.94	**0.00+**	5.90	**0.00+**	4.97	0.51	4.95	0.95
3	4.74	5.59	**0.00+**	5.62	**0.00+**	4.80	0.05	4.74	1.00
4	4.69	5.40	**0.00+**	5.41	**0.00+**	4.72	0.42	4.69	0.99
5	4.67	5.31	**0.00+**	5.29	**0.00+**	4.66	0.88	4.67	0.95
6	4.64	5.28	**0.00+**	5.28	**0.00+**	4.63	0.67	4.64	0.87
7	4.62	4.63	0.78	4.90	**0.00+**	4.60	0.62	4.62	0.98
8	4.57	4.57	1.00	4.57	1.00	4.57	1.00	4.57	1.00

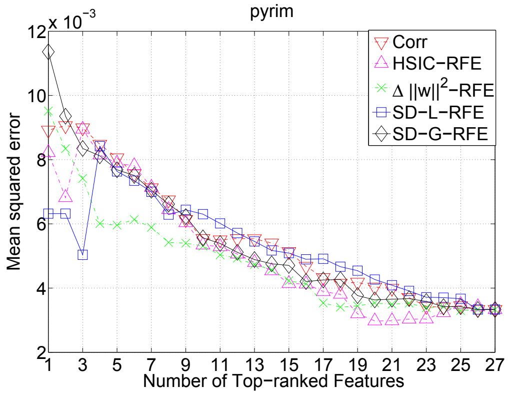

Figure 7: Average MSE against top-ranked features over 30 realizations for pyrim

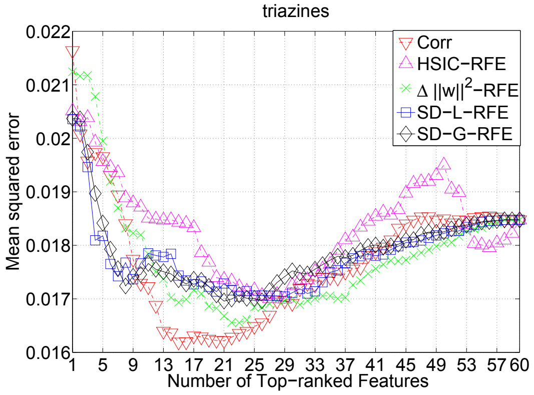

Figure 8: Average MSE against top-ranked features over 30 realizations for triazines

suggests that the proposed method generally performs as least as well, if not better than three benchmark methods, Corr, HSIC-RFE and $\Delta\|\omega\|^2$-RFE. 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.

7. REFERENCES

[1] A.Asuncion and D.J.Newman. UCI machine learning repository, 2007.

[2] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, November 1995.

[3] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001.

[4] W. Chu, S. S. Keerthi, and C. J. Ong. Bayesian support vector regression using a unified loss function. *IEEE Transactions on Neural Networks*, 15:29–44, 2004.

[5] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, November 2000.

[6] J. H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67, 1991.

[7] J. B. Gao, S. R. Gunn, C. J. Harris, and M. Brown. A probabilistic framework for SVM regression and error bar estimation. *Machine Learning*, 46(1-3):71–89, 2002.

[8] O. Gualdrón, J. Brezmes, E. Llobet, A. Amari, X. Vilanova, B. Bouchikhi, and X. Correig. Variable selection for support vector machine based multisensor systems. *Sensors and Actuators B: Chemical*, 122:259–268, March 2007.

[9] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[10] I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, editors. *Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing)*. Springer Verlag, August 2006.

[11] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

[12] T. Joachims. *Making large-Scale SVM Learning Practical.*, chapter In B. Scholkopf, C. Burges and A. Smola (Eds), Advances in kernel methods: Support Vector Learning. MIT Press, 1998.

[13] Y. Kim and J. Kim. Gradient lasso for feature selection. In *Proceedings of the 21st International Conference on Machine Learning*, pages 60–67, 2004.

[14] M. H. Law and J. T. Kwok. Bayesian support vector regression. In *In Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics*, pages 239–244, 2001.

[15] C. J. Lin and R. C. Weng. Simple probabilistic predictions for support vector regression. Technical report, Department of Cmputer Science, National Taiwan University, 2004.

[16] D. MacKay. The evidence framework applied to classification networks. *Neural Computation*, 4(5):720–736, 1992.

[17] A. Navot, L. Shpigelman, N. Tishby, and E. Vaadia. Nearest neighbor based feature selection for regression and its application to neural activity. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 995–1002, Cambridge, MA, 2006. MIT Press.

[18] A. Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *ICML '04: Proceedings of the twenty-first International Conference on Machine learning*, pages 78–85, New York, NY, USA, 2004. ACM.

[19] J. C. Platt. *Using sparseness and analytic QP to speed training of support vector machines*, chapter In M.S. Kearns, S.A. Solla and D. A. Cohn (Eds), Advances in Neural Information Processing Systems, 11. Cambridge, MIT Press, 1998.

[20] A. Rakotomamonjy. Variable selection using SVM-based criteria. *Journal of Machine Learning Research*, 3:1357–1320, 2003.

[21] K. Q. Shen, C. J. Ong, X. P. Li, and E. P. Wilder-Smith. Feature selection via sensitivity analysis of SVM probabilistic outputs. *Machine Learning*, 70(1):1–20, 2008.

[22] L. Song, A. Smola, A. Gretton, J. Bedo, and K. Borgwardt. Supervised feature selection via dependence estimation. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 823–830, New York, NY, USA, 2007. ACM.

[23] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 58(1):267–288, 1996.

[24] V. N. Vapnik. *Statistical Learning Theory.* Wiley-Interscience, September 1998.

[25] 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 Transactions on Neural Network*, 20(12):1911 – 1922, December 2009.