Model Combination for Support Vector Regression via Regularization Path

Mei Wang[1,2] and Shizhong Liao[1]

[1] School of Computer Science and Technology,
Tianjin University,Tianjin, China
[2] School of Computer and Information Technology,
Northeast Petroleum University, Daqing, China
szliao@tju.edu.cn

Abstract. In order to improve the generalization performance of support vector regression (SVR), we propose a novel model combination method for SVR on regularization path. First, we construct the initial candidate model set using the regularization path, whose inherent piecewise linearity makes the construction easy and effective. Then, we elaborately select the models for combination from the initial model set through the improved Occam's Window method and the input-dependent strategy. Finally, we carry out the combination on the selected models using the Bayesian model averaging. Experimental results on benchmark data sets show that our combination method has significant advantage over the model selection methods based on generalized cross validation (GCV) and Bayesian information criterion (BIC). The results also verify that the improved Occam's Window method and the input-dependent strategy can enhance the predictive performance of the combination model.

Keywords: Model combination, Support vector regression, Regularization path, Occam's Window.

1 Introduction

Support vector regression (SVR) [1] is an extension of the support vector method to regression problem, which maintains all the main characteristics of the maximal margin algorithm. The generalization performance of SVM depends on the parameters of regularization and kernels. Various algorithms [2,3] have been developed for choosing the best parameters. Regularization path algorithm is another important algorithm to address the SVR model selection problem [4,5], which can fit the entire path of SVR solutions for every value of the regularization parameter. Gunter and Zhu [4] proposed an unbiased estimate for the degrees of freedom of the SVR model, then applied the generalized cross validation (GCV) criterion [6] to select the optimal model. However, single model only has limited information and usually exists uncertainty [7,8]. Model combination is an alternative way to overcome the limitations of model selection,

P. Anthony, M. Ishizuka, and D. Lukose (Eds.): PRICAI 2012, LNAI 7458, pp. 649–660, 2012.

which can integrate all useful information from the candidate models into the final hypothesis to improve generalization performance. There are a lot of experimental works showing that combining learning machines often leads to improved generalization performance [9,10,11,12,13].

In this paper, we study the model combination for SVR on regularization path. First, the initial candidate model set is obtained according to the regularization path, whose inherent piecewise linearity makes the construction easy and effective. All possible models are involved in the initial model set, including good performance ones and bad performance ones. Then, a subset from all available individual SVR models is selected by the improved Occam's Window method and the input-dependent strategy. The improved Occam's Window method can eliminate the model with poor performance and select the sparse model. The input-dependent strategy can determine the combination model set according to the estimation of the generalization error of the input. Finally, The combination on the selected models is carried out using the Bayesian model averaging, in which the model posterior probability is estimated by Bayesian information criterion (BIC) approximation.

2 ϵ-SVR Regularization Path

In this section, we briefly introduce the ϵ-SVR regularization path algorithm and refer readers to [4] for a detailed tutorial. The training data set has been taken as $T = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)\} \subset \mathbb{R}^p \times \mathbb{R}$, where the input $\boldsymbol{x}_i$ is a vector with p predictor variables, and the output y_i denotes the response. In ϵ-SVR, our goal is to find a function

$$f(\boldsymbol{x}) = \beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x} \rangle, \text{ with } \quad \boldsymbol{\beta} \in \mathbb{R}^p, \ \beta_0 \in \mathbb{R},$$

that has at most ϵ deviation from the actually obtained targets y for all the training data, and at the same time is as flat as possible. In practice, one often maps $\boldsymbol{x}$ onto a high dimensional reproducing kernel Hilbert space (RKHS), and fits a nonlinear kernel SVR model. Using the following ϵ-insensitive loss function

$$|y - f(\boldsymbol{x})|_\epsilon = \begin{cases} 0, & \text{if } |y - f(\boldsymbol{x})| < \epsilon, \\ |y - f(\boldsymbol{x})| - \epsilon, & \text{otherwise,} \end{cases}$$

the standard *loss + penalty* criterion of the ϵ-SVR model may be written as

$$\min_{f \in \mathcal{H}_K} \sum_{i=1}^{n} |y_i - f(\boldsymbol{x}_i)|_\epsilon + \frac{\lambda}{2} \|f\|^2_{\mathcal{H}_K}, \tag{1}$$

where λ is the regularization parameter, and $\mathcal{H}_K$ is a structured RKHS generated by a positive definite kernel $K(\boldsymbol{x}, \boldsymbol{x}')$. Using the representer theorem [14], the solution to equation (1) has a finite form

$$f(\boldsymbol{x}) = \beta_0 + \frac{1}{\lambda} \sum_{i=1}^{n} \theta_i K(\boldsymbol{x}, \boldsymbol{x}_i), \text{ with } \quad \theta_i \in [-1, +1], \ i = 1, \ldots, n. \tag{2}$$

In this paper, we use $\boldsymbol{\theta}$ to denote the coefficient vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)^\top$.

According to the piecewise of the ϵ-insensitive loss function and the Karush-Kuhn-Tucker conditions, the training data set is partitioned into the following five disjoint sets:

$$\begin{aligned}
\mathcal{R} &= \{i : y_i - f(\boldsymbol{x}_i) > \epsilon,\ \theta_i = 1\},\\
\mathcal{E}_\mathcal{R} &= \{i : y_i - f(\boldsymbol{x}_i) = \epsilon,\ 0 \le \theta_i \le 1\},\\
\mathcal{C} &= \{i : -\epsilon < y_i - f(\boldsymbol{x}_i) < \epsilon,\ \theta_i = 0\},\\
\mathcal{E}_\mathcal{L} &= \{i : y_i - f(\boldsymbol{x}_i) = -\epsilon, -1 \le \theta_i \le 0\},\\
\mathcal{L} &= \{i : y_i - f(\boldsymbol{x}_i) < -\epsilon,\ \theta_i = -1\}.
\end{aligned}$$

The ϵ-SVR regularization path algorithm keeps track of the five sets, and examines these sets until one or both of them change. For example, a point from $\mathcal{C}$ enters $\mathcal{E}_\mathcal{R}$. Once there is a change in the elements of the sets, we will say an event has occurred and a breakpoint will appear on the regularization path. As a data point passes through $\mathcal{E}_\mathcal{R}$ or $\mathcal{E}_\mathcal{L}$, its respective θ_i must change from 1 to 0 or -1 to 0 or vice versa. The algorithm begins with $\lambda^0 = \infty$ and the initial sets $\mathcal{E}_\mathcal{R}$ and $\mathcal{E}_\mathcal{L}$ have at most one point combined. The initial solution is obtained by solving a linear programming problem. Then the algorithm recursively computes λ^l ($l \in \mathbb{N}$). Each λ^l corresponds to the value of λ when an event occurs. The λ^{l+1} will be the largest λ less than λ^l such that either θ_i ($i \in \mathcal{E}_\mathcal{R}^l$) reaches 0 or 1, or θ_j ($j \in \mathcal{E}_\mathcal{L}^l$) reaches 0 or -1, or one of the points in $\mathcal{R}$, $\mathcal{L}$, or $\mathcal{C}$ reaches an elbow. When λ^{l+1} is known, the index sets $\mathcal{R}$, $\mathcal{E}_\mathcal{R}$, $\mathcal{C}$, $\mathcal{E}_\mathcal{L}$, $\mathcal{L}$ and $\boldsymbol{\theta}$ are updated according to the nature of the transition that had taken place to yield $\mathcal{R}^{l+1}$, $\mathcal{E}_\mathcal{R}^{l+1}$, $\mathcal{C}^{l+1}$, $\mathcal{E}_\mathcal{L}^{l+1}$, $\mathcal{L}^{l+1}$ and $\boldsymbol{\theta}^{l+1}$. This main phase proceeds repeatedly in increasing value of l and decreasing value of λ^l starting from λ^0 until termination. It is worth noting that the θ_is ($i \in \mathcal{L} \cup \mathcal{R}$) do not change in value when no new event happens. The algorithm will be terminated either when the sets $\mathcal{R}$ and $\mathcal{L}$ become empty or when λ has become sufficiently close to zero.

The whole solution path $\boldsymbol{\theta}(\lambda)$ is piecewise linear. As long as the break points can be establish, all values in between can be found by simple linear interpolation. Figure 1 shows the paths of all the $\{\theta_i(\lambda) \mid 0 < \lambda < \infty\}$ for data set *pyrim* with $n = 7$.

3 Model Combination for SVR

In this section, we will present how to construct the candidate model set according to the ϵ-SVR regularization path and how to combine the models.

3.1 Initial Model Set Based on Regularization Path

As we have stated in the former section, the regularization path algorithm can compute the exact entire regularization path, which can facilitate the selection of a model. The path $\{\boldsymbol{\theta}(\lambda), 0 \le \lambda \le \infty\}$ ranges from the least regularized model to the most regularized model. We adopt the notation $f(\boldsymbol{x}; \boldsymbol{\theta}, \lambda)$ for a model with

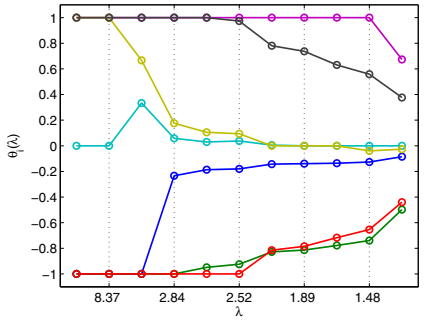

Fig. 1. The entire collection of piecewise liner paths $\theta_i(\lambda),\ i = 1, \ldots, n$ for the data set *pyrim*

parameter $\boldsymbol{\theta}$ and regularization parameter λ. It should be understood that the different models may be parameterized differently. Hence by $f(\boldsymbol{x}; \boldsymbol{\theta}, \lambda)$ we really mean $f(\boldsymbol{x}; \boldsymbol{\theta}(\lambda), \lambda)$ or $f_\lambda(\boldsymbol{x}; \boldsymbol{\theta})$, and we use the notation f_λ for simplicity.

The solution $\boldsymbol{\theta}(\lambda)$ is piecewise linear as a function of λ. We let the sequence $\infty > \lambda_1 > \cdots > \lambda_k > 0$ denote the corresponding break points on the path. In the interior of any interval $(\lambda_{l+1}, \lambda_l),\ 0 < l < k$, the set $\mathcal{L}, \mathcal{E}_\mathcal{L}, \mathcal{C}, \mathcal{E}_\mathcal{R}, \mathcal{R}$ are constant with respect to λ, such that the support vectors (i.e. the points with $\theta_i \neq 0$) remain unchanged. Therefore, all the regularization parameter in $(\lambda_{l+1}, \lambda_l)$ can lead to models with the same complexity. So we select $\boldsymbol{\theta}$s and λs on the break points to obtain the initial candidate model set $\mathcal{M}_{init} = \{f_{\lambda_1}, \ldots, f_{\lambda_k}\}$. The total number k of break points is $c \times n$, where n is the size of the training set and c is some small number around 1-6.

From the initial candidate ϵ-SVR model set, we can not directly perform the model combination over it, because most of the models in $\mathcal{M}_{init}$ are trivial in the sense that they predict the data far less well than the best models. So, we perform the model combination over a subset of parsimonious, data-supported models. In the latter subsections, we propose two simple and efficient ways of selecting models to guarantee the good performance of model combination.

3.2 Improved Occam's Window

All possible models are involved in the initial model set $\mathcal{M}_{init}$, including the good performance ones and the poor performance ones. Madigan and Raftery [15] used the Occam's Window method for graphical model and showed combination on the selected models provided better inference performance than basing inference on a single model in each of the examples they considered. In this paper, we apply an improved Occam's Window method to eliminate the poor performance models. In the proposed method, posterior model probabilities are used as a metric to guide model selection. There are two basic principles underlying this approach.

First, if a model predicts the data far less well than the model which provides the best prediction, then it has been discredited and should no longer be considered. Thus the model not belonging to should be excluded from the combination candidate model set, where posterior probability ratio W is chosen by the data analyst and $\max_l\{\Pr(f_{\lambda_l} \mid T)\}$ denotes the model in initial candidate model set $\mathcal{M}_{init}$ with the highest posterior model probability.

$$\mathcal{M}' = \left\{ f_{\lambda_j} : \frac{\max_l\{\Pr(f_{\lambda_l} \mid T)\}}{\Pr(f_{\lambda_j} \mid T)} \leq W \right\}$$

Secondly, appealing to Occam's razor, we exclude models which receive less support from the data than any of their simpler submodels. Here we give the definition of submodel. If f_{λ_i} is a submodel of f_{λ_j}, we mean that all the support vectors involved in f_{λ_i} are also in f_{λ_j}. Thus we also exclude from models belonging to then we obtain the model set $\mathcal{M}_{ao} = \mathcal{M}' \backslash \mathcal{M}'' \subseteq \mathcal{M}_{init}$.

$$\mathcal{M}'' = \left\{ f_{\lambda_j} : \exists f_{\lambda_l} \in \mathcal{M}_{init}, f_{\lambda_l} \subset f_{\lambda_j}, \frac{\Pr(f_{\lambda_l} \mid T)}{\Pr(f_{\lambda_j} \mid T)} > 1 \right\},$$

The posterior probability ratio W is usually a constant as in [15]. However, the statistical results show that only few of the ϵ-SVR models in $\mathcal{M}_{init}$ have strongly peak posterior probabilities, as shown in Figure 2. So, we apply a query-dependent method to determine the ratio W. Starting from $W = k/20$, we double it for every iteration and examine the number of models in set $\mathcal{M}''$. Once the number changes dramatically, we terminate the iteration process and use the last W value. In the experiments of the next section, this would be the case with the model set size $|\mathcal{M}'|$ increasing more than 4. Here, if the model set size enlarges dramatically, it means that many models with low posterior probabilities enter the model set $|\mathcal{M}'|$.

The improved Occam's Window algorithm, as shown in the Algorithm 1, can greatly reduce the number of models in the candidate model set. Typically, in our experience, the number of the candidate model set is reduced to fewer than $k/20$.

3.3 Input-Dependent Strategy

Though the improved Occam's Window method, we have obtained a credible candidate model set on the training data. Further, in order to perform good prediction on new input, we need a more credible input-dependent subset for model combination.

The generalization performance of the model combination can be evaluated by prediction error on the new input $\boldsymbol{x}$. For regression, we apply quadratic error $(f_{bma}(\boldsymbol{x}) - y)^2$ to calculate the prediction error of the model combination, where f_{bma} denotes the combined model.

Since the probability distribution according to which the data generated is unknown, it is impossible for us to compute the expectation of the combination error. In this paper, we use the nearest neighbor method to estimate the combination expected error on the input $\boldsymbol{x}$. Specifically, we adopt the search strategy

Algorithm 1. The improved Occam's Window algorithm.

Input: $\mathcal{M}_{init} = \{f_{\lambda_1}, \ldots, f_{\lambda_k}\}$, $\mathcal{P} = \{\Pr(f_{\lambda_1}), \ldots, \Pr(f_{\lambda_k})\}$, k, s
Output: $\mathcal{M}_{ao}$
$MP \leftarrow \max(\mathcal{P})$;
$\mathcal{M}_{ao} \leftarrow \emptyset$;
$\mathcal{M}_{tmp} \leftarrow \emptyset$;
$W \leftarrow k/20$;
while $\mathcal{M}_{init} \neq \emptyset$ **do**
  for $f \in \mathcal{M}_{init}$ **do**
    if $MP/\Pr(f) \leq W$ **then**
      $\mathcal{M}_{tmp} \leftarrow \mathcal{M}_{tmp} \cup \{f\}$;
      $\mathcal{M}_{init} \leftarrow \mathcal{M}_{init} \backslash \{f\}$
    end
  end
  if $|\mathcal{M}_{tmp}| - |\mathcal{M}_{ao}| \leq s$ **then**
    $\mathcal{M}_{ao} \leftarrow \mathcal{M}_{tmp}$;
    $W \leftarrow W * 2$;
  end
  else
    $\mathcal{M}_{ao} \leftarrow \mathcal{M}_{tmp}$;
    break
  end
end
for $f \in \mathcal{M}_{ao}$ **do**
  for $f_1 \in \mathcal{M}_{ao} \backslash \{f\}$ **do**
    if *(*$\Pr(f_1) > \Pr(f)$*) and (*$f_1 \subset f$*)* **then**
      $\mathcal{M}_{ao} \leftarrow \mathcal{M}_{ao} \backslash \{f\}$;
      break;
    end
  end
end

which computes the Euclidean distances between the input $\boldsymbol{x}$ and each point in the training set and then selects the one with smallest distance.

Suppose the input $\boldsymbol{x}$'s nearest neighbor we find is $\boldsymbol{x}_e$, $e \in [1, n]$ and its output is denoted by y_e. For each ϵ-SVR model $f_{\lambda_j} \in \mathcal{M}_{ao}$, we compute the prediction error $(f_{\lambda_j}(\boldsymbol{x}_e) - y_e)^2$, and sort them by ascending order. We add the model with current smallest error in $\mathcal{M}_{ao}$ to candidate model set, denoted by $\mathcal{M}_{aa}$, meanwhile remove it from $\mathcal{M}_{ao}$. Then we perform the model combination over $\mathcal{M}_{aa}$, and then compute the combination error $(f_{bma}(\boldsymbol{x}_e) - y_e)^2$. Until the combination error no longer declines, the model selection process will be terminated.

Since the whole combination process is dynamic, once a model is added to the model set $\mathcal{M}_{aa}$, we should update the posterior probabilities for each model. The model selection process is shown in Algorithm 2.

Algorithm 2. Input-dependant model selection algorithm.

Input: $\mathcal{M}_{ao}$, $T = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$, $\boldsymbol{x}$
Output: $\mathcal{M}_{aa}$
Find the nearest neighbor point $\boldsymbol{x}_e$ of $\boldsymbol{x}$ from T;
Compute the prediction error for each model on $\boldsymbol{x}_e$;
for $\mathcal{M}_{ao} \neq \emptyset$ **do**
  $f_m \leftarrow f \in \mathcal{M}_{ao}$ with lowest prediction error;
  $\mathcal{M}_{aa} \leftarrow \mathcal{M}_{aa} \cup \{f_m\}$;
  $\mathcal{M}_{ao} \leftarrow \mathcal{M}_{ao} \setminus \{f_m\}$;
  Update posterior probabilities of models in $\mathcal{M}_{aa}$;
  Compute $f_{bma}(\boldsymbol{x}_e)$;
  if $(f_{bma}(\boldsymbol{x}_e) - y_e)^2$ *is greater than last time* **then**
    break;
  end
end

3.4 Bayesian Model Averaging

For ϵ-SVR, we apply the model combination method—Bayesian model averaging. Suppose we have a ϵ-SVR candidate model set $\mathcal{M} = \{f_1, \ldots, f_m\}$. The Bayesian model averaging over $\mathcal{M}$ has the form

$$f_{bma}(\boldsymbol{x}) = \sum_{j=1}^{m} f_j(\boldsymbol{x}) \Pr(f_j \,|\, T), \tag{3}$$

where $\Pr(f_j \,|\, T)$ is the posterior probability of model f_j, $j = 1, \ldots, m$. Then this is the process of estimating the prediction under each model f_j and then averaging the estimates according to how likely each model is.

We can perform Bayesian model averaging over any ϵ-SVR model set, such as $\mathcal{M}_{init}$, $\mathcal{M}_{ao}$ and $\mathcal{M}_{aa}$, while for new input we should use the selected model set $\mathcal{M}_{aa}$.

In general, the posterior probability of model f_j in equation (3) is given by

$$\Pr(f_j \,|\, T) \propto \Pr(T \,|\, f_j)\Pr(f_j), \tag{4}$$

where $\Pr(T \,|\, f_j)$ is the marginal likelihood of model f_j and $\Pr(f_j)$ is the prior probability that f_j is the true model. In this paper, we will propose a simple and efficient method to estimate the model posterior probability for fixed regularization parameter λ, which depends on the ϵ-SVR regularization path algorithm. We estimate the posterior probability of each model f_j as [16]

$$\widehat{\Pr}(f_j \,|\, T) = \frac{e^{-\frac{1}{2}\cdot \mathrm{BIC}_j}}{\sum_{f_m \in \mathcal{M}} e^{-\frac{1}{2}\cdot \mathrm{BIC}_m}}, \tag{5}$$

and for each model f_j in model set $\mathcal{M}$, its BIC value can be calculated as

$$\mathrm{BIC}(f_j) = \frac{\|\boldsymbol{y} - \boldsymbol{f}_j\|^2}{n\sigma^2} + \frac{\log(n)}{n}\mathrm{df}(f_j), \tag{6}$$

where $\boldsymbol{y} = (y_1, \ldots, y_n)^\top$, $\boldsymbol{f}_j = (f_j(\boldsymbol{x}_1), \ldots, f_j(\boldsymbol{x}_n))^\top$, $j = 1, \ldots, k$.

3.5 Computational Complexity

The main computational burden of the model combination for ϵ-SVR centers on building the ϵ-SVR regularization path, proceeding improved Occam's Window procedure to exclude models to obtain the model set $\mathcal{M}_{ao}$, and selecting models using the input-dependant strategy to obtain the model set $\mathcal{M}_{aa}$.

The approximate computational complexity of the ϵ-SVR regularization path algorithm is $O(cn^2m+nm^2)$ [4], where n is the size of the training data and m is the average size of $\mathcal{E}_{\mathcal{R}} \cup \mathcal{E}_{\mathcal{L}}$, and c is some small number as previously mentioned.

The search strategy in the improved Occam's Window method is to identify the models in $\mathcal{M}_{ao}$. First part of the method involves $O(dk)$ operations, including finding the largest posterior probability and excluding the models with low posterior probability. Here, $k = c \times n$ is the size of the initial candidate model set, and d is the iteration for adjusting W, our experience so far suggests that d is around $3-8$. Second part of the method involves $O(k^2m)$ operations, including determining subset relationship and comparing the posterior probabilities between each pair of the models, and the m is the average size of $\mathcal{E}_{\mathcal{R}} \cup \mathcal{E}_{\mathcal{L}}$. So the approximate computational complexity of the improved Occam's Window is $O(cn^2m)$.

The approximate computational complexity of the input-dependant strategy is $O(cn)$, including finding the nearest neighbor data point from the training data and determining the final candidate model set for combination.

So, the total computational complexity of the model combination for ϵ-SVR on regularization path is $O(cn^2m + nm^2)$.

4 Experiments

In this section, we investigate the performance of our model combination with GCV-based and BIC-based model selection on seven benchmark data sets used in [2] (available online at http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/), and we consider Gaussian radial basis kernel $K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\gamma\|\boldsymbol{x}-\boldsymbol{x}'\|^2)$, where γ is the prespecified kernel parameter. We use the same values for γ and ϵ as specified in [2] shown in Table 1, where n denotes the size of the data set, and p denotes the dimension of the input $\boldsymbol{x}$.

For data set *abalone* we randomly sample 1000 examples from the 4177 examples; for *cpusmall* we randomly sample 1000 examples from the 8292 examples; for *spacega* we randomly sample 1000 examples from the 3107 examples.

Since the usual goal of regression analysis is to minimize the predicted squared-error loss, the prediction error is defined as

$$\text{PredE} = \frac{1}{m}\sum_{i=1}^{m}(y_i - f(\boldsymbol{x}_i))^2,$$

where m is the number of the test data.

Table 1. Summary of the Seven Benchmark Data Sets

DataSet	n	p	γ	ϵ	DataSet	n	p	γ	ϵ
pyrim	74	27	0.0167	0.00136	cpusmall	1000	12	0.0913	0.12246
triazines	186	60	0.0092	0.00910	spacega	1000	6	0.1664	0.01005
mpg	392	7	0.3352	0.18268	abalone	1000	8	0.1506	0.12246
housing	566	13	0.1233	0.18268					

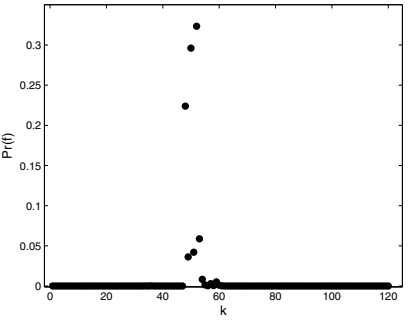

Fig. 2. The posterior probability distribution of the models according to the regularization path

4.1 Model Posterior Probability Distribution

First, we verify the model posterior probabilities according to the regularization path. We randomly sample on data set *pyrim*, and then build each regularization path using the algorithm proposed in [4]. We compute the model posterior probability as described in Section 3. The posterior probability of models from most regularized to least regularized is shown in Figure 2.

From the figure we find that only few models have higher posterior probabilities and most of the other models have very small posterior probabilities around zero. Therefore, applying the improved Occam's Window method we can discard most of the models in the initial candidate model set. We record an example of adjusting procedure on posterior probability ratio W in Table 2, where the ATs is the iteration on the W. We observe that once the ratio is large enough, many models enter the candidate model set. From the last line of the table, we can conclude that poor performance models can decrease the performance of model combination.

4.2 Performance Comparison

In this subsection, we first compare the prediction performance of the model combination over $\mathcal{M}_{aa}$ with the model selection methods based GCV [4] and

Table 2. Adjusting procedure on posterior probability ratio W with data set *pyrim*

Ats	W	$\lvert\mathcal{M}'\rvert$	PredE
1	6	2	0.00593
2	12	5	0.00601
3	24	7	0.00596
4	48	9	0.00570
5	96	22	0.00713

BIC. We randomly split the data into training and test sets, with the training set comprising 80% of the data. We repeat this process 30 times and compute the average prediction errors and their corresponding standard errors. We calculate the prediction error with each test data for each method. The results are summarized in Table 3. From the tables we find that the model combination has the lowest prediction error and standard error. In a sense, this experiment shows that model combination has the property of "many can be better than one".

Table 3. Comparisons of the prediction error on real data for model selection and model combination

	GCV	BIC	BMC
pyrim	0.0055 (0.0026)	0.0052 (0.0027)	0.0049 (0.0023)
triazines	0.0242 (0.0081)	0.0239 (0.0080)	0.0237 (0.0078)
mpg	7.32 (2.35)	7.25 (2.32)	7.13 (2.12)
housing	10.82 (3.65)	10.77 (3.60)	10.04 (3.49)
cpusmall	27.48 (10.25)	27.32 (10.25)	27.10 (10.23)
spacega	0.0125 (0.0015)	0.0122 (0.0015)	0.0120 (0.0015)
abalone	4.31 (1.05)	4.29 (1.05)	4.27 (1.04)

In the second part of the experiment, we compare the prediction performance of model combination over the model set $\mathcal{M}_{init}$, $\mathcal{M}_{ao}$ and $\mathcal{M}_{aa}$. We compute the prediction error for the model combination with and without the model selection strategy. The results are summarized in Table 4, where BMC_i denotes the model combination over the initial candidate model set $\mathcal{M}_{init}$; BMC_o denotes combination over the model set $\mathcal{M}_{ao}$, and BMC_p denotes combination over the final model set $\mathcal{M}_{aa}$. From the tables we find that the model combination over the selected candidate model set has lower prediction error than over the initial model set. In a sense, this experiment shows that the model combination has the property of "many can be better than all".

Table 4. Comparisons of the prediction error on real data for model combination with and without model selection strategy

	BMC_i	BMC_o	BMC_p
pyrim	0.0067 (0.0037)	0.0051 (0.0027)	0.0049 (0.0023)
triazines	0.0317 (0.0092)	0.0277 (0.0083)	0.0237 (0.0078)
mpg	7.98 (2.68)	7.28 (2.32)	7.13 (2.12)
housing	11.73 (3.98)	10.73 (3.56)	10.04 (3.49)
cpusmall	29.49 (11.08)	28.01 (10.78)	27.10 (10.23)
abalone	4.83 (1.27)	4.36 (1.09)	4.27 (1.04)
spacega	0.0204 (0.0019)	0.0128 (0.0015)	0.0120 (0.0015)

5 Conclusion

In this paper, we propose a new model combination framework for ϵ-SVR. We can obtain all possible models according to the regularization path. Applying the improved Occam's Window method and the input-dependant strategy, we greatly reduce the number of candidate models and improve the model combination prediction performance on test data. The model combination on regularization path can reduce the risk of single model selection, and improve the prediction performance. The experimental results on real data show that with some pre-processing of model set the combination prediction accuracy significantly exceeds that of a single model.

Our model combination for ϵ-SVR on regularization path provide a common framework for model combination, which can be extended to support vector machines (SVMs)[17] and other regularized models.

Acknowledgment. The authors thank the three referees for their helpful comments. This work is supported by Natural Science Foundation of China under Grant No. 61170019 and Natural Science Foundation of Tianjin under Grant No. 11JCY BJC00700.

References

1. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Statistics and Computing 14(3), 199–222 (2004)
2. Chang, M.W., Lin, C.J.: Leave-one-out bounds for support vector regression model selection. Neural Computation 17(5), 1188–1222 (2005)
3. Wang, G., Yeung, D., Lochovsky, F.: Two-dimensional solution path for support vector regression. In: Proceedings of the 23th International Conference on Machine Learning, pp. 993–1000 (2006)
4. Gunter, L., Zhu, J.: Efficient computation and model selection for the support vector regression. Neural Computation 19(6), 1633–1655 (2007)

5. Wang, G., Yeung, D.Y., Lochovsky, F.H.: A new solution path algorithm in support vector regression. IEEE Transactions on Neural Networks 19(10), 1753–1767 (2008)
6. Craven, P., Wahba, G.: Smoothing noisy data with spline functions. Numerische Mathematik 31(4), 377–403 (1978)
7. Petridis, V., Kehagias, A., Petrou, L., Bakirtzis, A., Kiartzis, S., Panagiotou, H., Maslaris, N.: A bayesian multiple models combination method for time series prediction. Journal of Intelligent and Robotic Systems 31(1), 69–89 (2001)
8. Freund, Y., Mansour, Y., Schapire, R.E.: Why averaging classifiers can protect against overfitting. In: Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics, vol. 304. Citeseer (2001)
9. Ji, C., Ma, S.: Combinations of weak classifiers. IEEE Transactions on Neural Networks 8(1), 32–42 (1997)
10. Raftery, A.E., Madigan, D., Hoeting, J.A.: Bayesian model averaging for linear regression models. Journal of the American Statistical Association 92, 179–191 (1997)
11. Kittler, J.: Combining classifiers: A theoretical framework. Pattern Analysis and Applications 1, 18–27 (1998)
12. Evgeniou, T., Pontil, M., Elisseeff, A.: Leave one out error, stability, and generalization of voting combinations of classifiers. Machine Learning 55(1), 71–97 (2004)
13. Bagui, S.C.: Combining pattern classifiers: methods and algorithms. Technometrics 47(4), 517–518 (2005)
14. Kimeldorf, G., Wahba, G.: Some results on Tchebycheffian spline functions. Journal of Mathematical Analysis and Applications 33(1), 82–95 (1971)
15. Madigan, D., Raftery, A.E.: Model selection and accounting for model uncertainty in graphical models using Occam's window. Journal of the American Statistical Association 89(428), 1535–1546 (1994)
16. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer (2008)
17. Zhao, N., Zhao, Z., Liao, S.: Probabilistic Model Combination for Support Vector Machine Using Positive-Definite Kernel-Based Regularization Path. In: Wang, Y., Li, T. (eds.) ISKE2011. AISC, vol. 122, pp. 201–206. Springer, Heidelberg (2011)