# Non-monotonic Feature Selection for Regression

Haiqin Yang[1,2], Zenglin Xu[3,4], Irwin King[1,2], and Michael R. Lyu[1,2]

[1] Shenzhen Key Laboratory of Rich Media Big Data Analytics and Applications
Shenzhen Research Institute, The Chinese University of Hong Kong
[2] Computer Science & Engineering, The Chinese University of Hong Kong, Hong Kong
{hqyang,king,lyu}@cse.cuhk.edu.hk
[3] University of Electronic Science & Technology of China, Chengdu, Sichuan, China
[4] Purdue University, West Lafayette, IN, USA
zenglin@gmail.com

**Abstract.** Feature selection is an important research problem in machine learning and data mining. It is usually constrained by the budget of the feature subset size in practical applications. When the budget changes, the ranks of features in the selected feature subsets may also change due to nonlinear cost functions for acquisition of features. This property is called non-monotonic feature selection. In this paper, we focus on non-monotonic selection of features for regression tasks and approximate the original combinatorial optimization problem by a Multiple Kernel Learning (MKL) problem and show the performance guarantee for the derived solution when compared to the global optimal solution for the combinatorial optimization problem. We conduct detailed experiments to demonstrate the effectiveness of the proposed method. The empirical results indicate the promising performance of the proposed framework compared with several state-of-the-art approaches for feature selection.

## 1 Introduction

Feature selection is an important task in machine learning and data mining. The goal of feature selection is to choose a subset of informative features from the input data so as to reduce the computational cost or save storage space for problems with high dimensional data. Feature selection has found applications in a number of real-world problems, such as data visualization, natural language processing, computer vision, speech processing, bioinformatics, sensor networks and so on [10]. More information can be found from the comprehensive survey paper [4] and references therein.

A general definition of selecting features from a learning task is to choose a subset of $m$ features, denoted by $\mathcal{S}$, that maximizes a generalized performance criterion $\mathcal{Q}$. It is cast into the following combinatorial optimization problem:

$$\mathcal{S}^* = \arg\max \mathcal{Q}(\mathcal{S}) \quad \text{s. t.} \quad |\mathcal{S}| = m. \tag{1}$$

Here $m$ is also called the budget of selected features. $\mathcal{Q}(\mathcal{S})$ is restricted to a performance measure for regression problems. More specifically, we adopt the dual objective function of Support Vector Regression (SVR), a popular regression model [7] in the literature, as is defined later in Eq. (3).

C.K. Loo et al. (Eds.): ICONIP 2014, Part II, LNCS 8835, pp. 44–51, 2014.

When the budget changes, the new feature subset may not be a subset or superset of the previous feature subset due to nonlinear cost functions for acquisition of features. This property is called non-monotonic feature selection [12]:

**Definition 1 (Non-monotonic Feature Selection).** *A feature selection algorithm $\mathcal{A}$ is monotonic if and only if it satisfies the following property: for any two different numbers of selected features, i.e., $k$ and $m$, we always have $\mathcal{S}_k \subseteq \mathcal{S}_m$ if $k \leq m$, where $\mathcal{S}_m$ stands for the subset of $m$ features selected by $\mathcal{A}$. Otherwise, it is called non-monotonic feature selection.*

Due to the dependance of feature selection on the budget of feature subsets, traditional feature selection methods may yield sub-optimal solutions. In order to tackle this problem, in this paper, we propose a **non-monotonic** feature selection for regression. Following the framework for classification derived in [8,12], we approximate the original combinatorial optimization problem of feature selection and formulate it closely related to multiple kernel learning (MKL) [1,6,11,13,14,17] framework, which yields the final optimization problem to be solved efficiently by a Quadratically Constrained Quadratic Programming (QCQP) problem. Differently, support vector regression [7] is selected as the regression model due to its power in solving real-world applications. We then present a strategy that selects a subset of features based on the solution of the relaxed problem and show the **performance guarantee**, which bounds the difference in the value of objective function between using the features selected by the proposed strategy and using the global optimal subset of features found by exhaustive search. Our empirical study shows that the proposed approach performs better than the state-of-the-arts for feature selection in tackling the regression task.

## 2 Model and Analysis

Suppose the training set includes $N$ samples: $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^d$ represents the features of the $i$-th sample, and $y_i \in \mathbb{R}$ corresponds to the response. Let $\mathbf{e}_d \in \mathbf{R}^d$ be a $d$-dimensional vector with all elements being one and $\mathbf{I}_d$ be the $d \times d$ identity matrix. For a linear kernel, the kernel matrix $\mathbf{K}$ is written as: $\mathbf{K} = \mathbf{X}^\top \mathbf{X} = \sum_{i=1}^d \mathbf{x}_i \mathbf{x}_i^\top = \sum_{i=1}^d \mathbf{K}_i$, where a kernel $\mathbf{K}_i = \mathbf{x}_i \mathbf{x}_i^\top$ is defined for each feature. The goal of feature selection is to select a subset of $m < d$ features, i.e., to determine the value of $\mathbf{p}$ in the following form:

$$\mathbf{K}(\mathbf{p}) = \sum_{i=1}^{d} p_i \mathbf{x}_i \mathbf{x}_i^\top = \sum_{i=1}^{d} p_i \mathbf{K}_i, \tag{2}$$

where $p_i \in \{0, 1\}$ is a binary variable that indicates if the $i$th feature is selected, and $\mathbf{p} = (p_1, \ldots, p_d)$. As revealed in (2), to select $m$ features, we need to find optimal binary weights $p_i$ to combine the kernels derived from individual features. This observation motivates us to cast the feature selection problem into a multiple kernel learning problem.

Following the maximum margin framework with $\varepsilon$-insensitive loss function for support vector regression [7,15,16] and the derivation in [12], given a kernel matrix $\mathbf{K}(\mathbf{p}) = \sum_{i=1}^{d} p_i \mathbf{K}_i$, the regression model, $f(\mathbf{x}) = \sum_{i=1}^{d} w_i \mathbf{x}_i + b = \sum_{j=1}^{N} (\alpha_j - \alpha_j^*) \mathbf{K}(\mathbf{p})$, is found by solving the following optimization problem:

$$\tilde{\omega}(\mathbf{p}) := \begin{cases} \max\limits_{\beta} 2\mathbf{v}^\top \beta - \beta^\top \mathbf{Q}(\mathbf{p}) \beta \\ \text{s.t.} \quad 0 \le \beta \le C, \quad \mathbf{u}^\top \beta = 0 \end{cases} \tag{3}$$

where the variable $\beta = [\alpha; \alpha^*] \in \mathbb{R}^{2N}$, and $\alpha,\ \alpha^* \in \mathbb{R}^N$ are corresponding Lagrange multipliers used to push and pull $f(\mathbf{x})$ towards the outcome of $y$, respectively. $b$ corresponds to the dual variable of $\mathbf{u}^\top \beta = 0$. The linear coefficient $\mathbf{v}$ is defined as $[\mathbf{v}_1; \mathbf{v}_2]$, where $\mathbf{v}_1 = [-\varepsilon \mathbf{e}_N + \mathbf{y}]$ and $\mathbf{v}_2 = [-\varepsilon \mathbf{e}_N - \mathbf{y}]$. $\mathbf{u}$ in the equality constraint is defined as $[\mathbf{e}_N^\top, -\mathbf{e}_N^\top]$. The matrix $\mathbf{Q}(\mathbf{p}) = [\mathbf{K}(\mathbf{p}), -\mathbf{K}(\mathbf{p}); -\mathbf{K}(\mathbf{p}), \mathbf{K}(\mathbf{p})] \in \mathbb{R}^{2N \times 2N}$.

By approximating the indicator vector $\mathbf{p}$ to a continuous indicator, we have to solve the following optimization problem:

$$\min_{0 \le \mathbf{p} \le 1} \quad \tilde{\omega}(\mathbf{p}) \quad \text{s.t.} \quad \mathbf{p}^\top \mathbf{e}_d = m. \tag{4}$$

Following the derivation in [6,12], we can transform (3) to the following optimization problem:

$$\begin{aligned} \min_{\mathbf{p}, t, \nu, \delta, \theta} \quad & t + 2C\delta^\top \mathbf{e}_{2N} \qquad (5) \\ \text{s.t.} \quad & \begin{pmatrix} \mathbf{Q}(\mathbf{p}) & \mathbf{v} + \nu - \delta + \theta \mathbf{u} \\ (\mathbf{v} + \nu - \delta + \theta \mathbf{u})^\top & t \end{pmatrix} \succeq 0, \\ & \nu \ge 0,\ \delta \ge 0,\ \mathbf{p}^\top \mathbf{e}_d = m,\ 0 \le \mathbf{p} \le 1. \end{aligned}$$

To further speedup the semi-definite programming (SDP) problem in (5), we show in the following theorem that (5) can be reformulated into a Quadratically Constrained Quadratic Programming (QCQP) problem similar to [12]:

**Theorem 1.** *The optimization problem in (4) can be reduced to the following optimization problem:*

$$\begin{aligned} \max_{\alpha, \alpha^*, \lambda, \gamma} \quad & 2(\mathbf{v}_1^\top \alpha + \mathbf{v}_2^\top \alpha^*) - m\lambda - \gamma^\top \mathbf{e}_N \qquad (6) \\ \text{s.t.} \quad & \mathbf{e}_N^\top (\alpha - \alpha^*) = 0,\ \ 0 \le \alpha, \alpha^* \le C, \\ & (\alpha - \alpha^*)^\top \mathbf{K}_i (\alpha - \alpha^*) \le \lambda + \gamma_i,\ \ \gamma_i \ge 0. \end{aligned}$$

*The KKT conditions are*

$$\begin{aligned} & \mathbf{K}(\mathbf{p})(\alpha - \alpha^*) = \mathbf{v} + \nu - \delta + \theta \mathbf{u}, \\ & t = [\alpha; \alpha^*]^\top (\mathbf{v} + \nu - \delta + \theta \mathbf{u}), \\ & [\alpha; \alpha^*] \circ \nu = \mathbf{0},\ [\alpha; \alpha^*] \circ \delta = C\delta,\ \gamma \circ (\mathbf{e}_d - \mathbf{p}) = \mathbf{0}, \\ & p_i(\lambda + \gamma_i - (\alpha - \alpha^*)^\top \mathbf{K}_i (\alpha - \alpha^*) = 0,\ \ i = 1, \ldots, d. \end{aligned} \tag{7}$$

Now, by observing (7), we rank the features in the descending order of $\tau_i = (\alpha - \alpha^*)^\top \mathbf{K}_i (\alpha - \alpha^*) = (\sum_{j=1}^{N} X_{i,j} (\alpha_j - \alpha_j^*))^2 = w_i^2$, where $w_i$ is the weight computed for the $i$-th feature. We denote by $i_1, \dots, i_d$ the ranked features, and by $k_{\min}$ and $k_{\max}$ the smallest and the largest indices such that $\tau_{i_k} = \tau_{i_m}$ for $1 \leq k \leq d$. We divide features into three sets and derive the properties of $\lambda$ and $\mathbf{p}$ as follows:

$$\mathcal{A} = \{i_k | 1 \leq k < k_{\min}\},\ \mathcal{B} = \{i_k | k_{\min} \leq k \leq k_{\max}\},\ \mathcal{C} = \{i_k | k_{\max} < k \leq d\}. \quad (8)$$

**Corollary 1.** *We have the following properties for* $\lambda$ *and* $\mathbf{p}$*:*

$$\lambda \in [\tau_{1+k_{\max}}, \tau_m], \quad p_i = \begin{cases} 1,\ i \in \mathcal{A}, \\ 0,\ i \in \mathcal{C}. \end{cases} \quad (9)$$

*Remark.* The above corollary can be derived by analyzing the KKT conditions in (7). We then can conduct our non-monotonic feature selection for regression, namely **NMMKLR**, by the following three steps: 1) Solve $\alpha, \alpha^*$ in (6) by a linear objective function with quadratic constraints, a special case of the QCQP; 2) Compute $\tau_i = (\sum_{j=1}^{N} X_{i,j}(\alpha_j - \alpha_j^*))^2$; 3) Select the first $m$ features with the largest $\tau_i$.

Moreover, we provide the following theorem to show that the performance guarantee of the discrete solution constructed by our proposed algorithm and the combinatorial optimization problem in the form of (4).

**Theorem 2.** *The discrete solution constructed by our NMMKLR, denoted by* $\mathbf{p}$*, has the following performance guarantee for the combinatorial optimization problem defined in (1):* $\frac{\omega(\mathbf{p})}{\omega(\tilde{\mathbf{p}}^*)} \leq \frac{1}{1 - \sigma_{\max}(\mathbf{R}^{-1/2}\mathbf{B}\mathbf{R}^{-1/2})}$*, where* $\mathbf{R} = \mathbf{Q}(\mathbf{p}^*), \quad \mathbf{B} = \sum_{j \in \mathcal{B}} p_j^* \mathbf{K}_j$ *. The operator* $\sigma_{\max}(\cdot)$ *calculates the largest eigenvalue.* $\mathbf{p}^*$ *and* $\widetilde{\mathbf{p}}^*$ *denote the optimal solution of the relaxed optimization problem in (4) and the global optimal solution of the original combinatorial optimization problem defined in (1), respectively.*

The proof can be found in the long version of this paper. Theorem 2 indicates that by incorporating the required number of selected features, the resulting approximate solution could be more accurate than without it, which implies that the proposed NMMKLR algorithm produces a better approximation to the underlying combinatorial optimization problem (1).

## 3 Experiments

We conduct detailed experiments on both synthetic and real-world datasets and compare our proposed **NMMKLR** with the following state-of-the-art methods: 1) **Stepwise**: the forward stepwise feature selection method [2,3,4] [1]; 2) **SVR-LW**: features are selected with the largest absolute weights $|w_i|$ computed by SVR [7]; 3) **LASSO-LW**: features are selected with the largest absolute weights $|w_i|$ computed by LASSO [9].

We adopt the following two metrics to measure the model performance: 1) **Mean Square Error (MSE)**: $MSE = \sum_{i=1}^{N}(y_i - \hat{y}_i)^2/N$, which measures the discrepancy of the predictive response and real response; 2) $Q^2$ **statistics**: $Q^2 = \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{(y_i - \bar{y}_i)^2}$, which is scaled by the variance of the response. where $\hat{y}_i$ is the prediction of $y_i$ for the $i$-th test sample, and $\bar{y}$ is the mean of the actual response. Obviously, in both metrics, the smaller the corresponding value is, the better the performance is.

[1] `http://www.robots.ox.ac.uk/~parg/software/fsbox_1_0.tar`

*Experiments on synthetic dataset.* We first generate a toy dataset consisting of $d$(=12) dimensions by an additive model similar to that in [2]: $y_i = \sum_{j=1}^{4} jx_{ji} + e^{x_{5i}}$, where $y_i$ denotes the response for the $i$-th sample and $\mathbf{x}_j$ denotes the $j$-th feature, for $j = 1, \ldots, 12$. $x_{ji}$ denotes the element of the $j$-th feature on the $i$-th sample. Here, only the first five features contribute to the response and each of them is generated from an independently and identically distributed normal distribution. The rest 7 features are generated as follows: the 6-th feature is $\mathbf{x}_6 = \mathbf{x}_1 + 1$, which is correlated to $\mathbf{x}_1$; the 7-th feature is $\mathbf{x}_7 = \mathbf{x}_2 \circ \mathbf{x}_3$, which is the element-wise product of $\mathbf{x}_2$ and $\mathbf{x}_3$; the rest five features, i.e., $\mathbf{x}_8, \ldots, \mathbf{x}_{12}$, generated by standard normal distribution, are totally irrelevant to the response $y_i$. For convenience, we also denote the irrelevant features by $NV_1, \ldots, NV_5$, respectively.

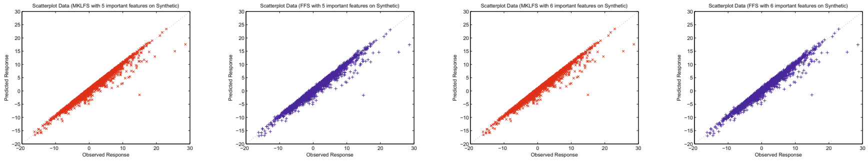
**Table 1.** Top 5 and 6 selected features (ordered) from four compared algorithms within 20 trials on the synthetic dataset. The stepwise and the LASSO-LW method include some irrelevant features when the number of selected features is greater than six.

| Method | Times | Top 5 selected features | Times | Top 6 selected features |
|---|---|---|---|---|
| NMMKLR | 19 | 4, 3, 2, 5, 1 | 20 | 4, 3, 2, 5, 1, 6 |
| | 1 | 4, 3, 2, 5, 6 | | |
| Stepwise | | | 8 | 4, 3, 2, 5, 1 |
| | 10 | 4, 3, 2, 5, 1 | 3 | 4, 3, 5, 2, 6 |
| | | | 2 | 4, 3, 2, 5, 6, 8 $(NV_1)$ |
| | 6 | 4, 3, 2, 5, 6 | 1 | 3, 4, 2, 5, 6 |
| | | | 1 | 4, 3, 2, 5, 6 |
| | 2 | 4, 3, 5, 2, 6 | 1 | 4, 3, 2, 5, 1, 9 $(NV_2)$ |
| | | | 1 | 4, 3, 2, 5, 1, 11 $(NV_4)$ |
| | 2 | 3, 4, 2, 6, 5 | 1 | 4, 3, 2, 5, 6, 9 $(NV_2)$ |
| | | | 1 | 4, 3, 2, 5, 6, 10 $(NV_3)$ |
| | | | 1 | 4, 3, 2, 5, 6, 12 $(NV_5)$ |
| SVR-LW | 10 | 4, 3, 2, 5, 1 | 10 | 4, 3, 2, 5, 1, 6 |
| | 10 | 4, 3, 2, 5, 6 | 10 | 4, 3, 2, 5, 6, 1 |
| LASSO-LW | | | 4 | 4, 3, 2, 5, 1, 8 $(NV_1)$ |
| | | | 3 | 4, 3, 2, 5, 1, 10 $(NV_3)$ |
| | 15 | 4, 3, 2, 5, 1 | 3 | 4, 3, 2, 5, 1, 12 $(NV_5)$ |
| | | | 2 | 4, 3, 2, 5, 1, 9 $(NV_2)$ |
| | | | 2 | 4, 3, 2, 5, 1, 11 $(NV_4)$ |
| | 4 | 4, 3, 2, 5, 6 | 1 | 4, 3, 2, 5, 1, 7 |
| | | | 1 | 4, 3, 2, 5, 6, 8 $(NV_1)$ |
| | | | 1 | 4, 3, 2, 5, 6, 9 $(NV_2)$ |
| | | | 1 | 4, 3, 2, 5, 6, 10 $(NV_3)$ |
| | 1 | 4, 3, 2, 6, 5 | 1 | 4, 3, 2, 5, 6, 11 $(NV_4)$ |
| | | | 1 | 4, 3, 2, 6, 5, 12 $(NV_5)$ |

We conduct two batches of experiments, where the budget is set to 5 and 6, respectively. Then in each batch of experiments, we randomly generate 200 samples and hold out 50% of the samples for training while keeping the rest for test. Each exepriment is then repeated 20 times.

In order to examine the property of the selected features, we list the top $5$ and $6$ selected features returned for all algorithms in Table 1. Obviously, our method can stably select those important features while SVR-LW also selects features relatively stable. However, the selected features by the forward stepwise feature selection method and the LASSO-LW method are unstable, and some irrelevant features are included when the number of selected features is greater than 5.

To further evaluate the regression performance on the selected features, we employ Support Vector Regression (SVR) as the regression model. We tune the hyperparameters $C$ and $\varepsilon$, of SVR through five-fold cross validation on the training data with the top $5$, the top $6$, and all the features. The hyperparameter of SVR, $C$, is chosen uniformly from the interval $[10^0, 10^3]$ on a logarithmic scale and $\varepsilon$ is chosen in $[0.01, 0.1, 0.25, 0.5, 0.75, 1, 1.5, 2]$.



(a) *NMMKLR*, $\#SF = 5$ (b) *Stepwise*, $\#SF = 5$ (c) *NMMKLR*, $\#SF = 6$ (d) *Stepwise*, $\#SF = 6$

**Fig. 1.** Scatter plots of the pairs $(\mathbf{y}, \hat{\mathbf{y}})$. (a) and (c) show the plot of NMMKLR when the number of selected features equal to 5 and 6, respectively. (b) and (d) shows the plot of Stepwise regression when the number of selected features equal to 5 and 6, respectively. It can be observed that (a) is thinner than (b) and (c) is thinner than (d).

Finally, we show the evaluation results on four compared algorithms in Table 2. It can be observed that the proposed NMMKLR outperforms other three methods in both of the $MSE$ and $Q^2$ measures in all cases. Moreover, the paired $t$-test with the confidence level of $95\%$ indicates that the advantage of NMMKLR is significant. To better visualize the difference between the response values predicted by the feature selection algorithms, we plot the pairs of observed response and predicted response, i.e., $(\mathbf{y}, \hat{\mathbf{y}})$, for the NMMKLR and the Stepwise selection method. The results are shown in Figure $1$. Ideally, if the $MSE$ is zero, all the points should drop on the line $\mathbf{y} = \hat{\mathbf{y}}$. Thus a scatter plot with smaller areas will be better. We observe from Figure 1 that the proposed NMMKLR has a better performance in both cases.

*Experiments on a real-world benchmark dataset.* We employ a real-world benchmark dataset, the Boston Housing problem [5] to evaluate the above four feature selection algorithms. The Boston Housing problem [5] is a popular benchmark dataset for evaulating regression models. It consists of $506$ instances with $13$ continuous features, such as crime rate, lower status of the population, etc. The response variable is the median value of owner-occupied homes in $\$1000$'s,

In the experiment, we normalize the continuous features in the range of $[-1, 1]$ and hold out half of samples for training while keeping the rest for test. The parameters of SVRs are tuned on the training data with the top $5$, the top $6$, and all features, where $C$

**Table 2.** The test accuracy for the synthetic dataset evaluated by the performance measures ($MSE$ and $Q^2$) on four algorithms. The best results are highlighted (achieved by the paired $t$-test with 95% confidence level).

| #SF | NMMKLR | | Stepwise | |
|---|---|---|---|---|
| | $MSE$ | $Q^2$ | $MSE$ | $Q^2$ |
| 5 | $\mathbf{1.1599} \pm 0.6977$ | $\mathbf{0.0339} \pm 0.0186$ | $1.2156 \pm 0.6893$ | $0.0356 \pm 0.0183$ |
| 6 | $\mathbf{1.1600} \pm 0.6977$ | $\mathbf{0.0339} \pm 0.0186$ | $1.2352 \pm 0.6787$ | $0.0362 \pm 0.0180$ |
| #SF | SVR-LW | | LASSO-LW | |
| | $MSE$ | $Q^2$ | $MSE$ | $Q^2$ |
| 5 | $1.2128 \pm 0.7421$ | $0.0353 \pm 0.0198$ | $1.2156 \pm 0.6893$ | $0.0356 \pm 0.0183$ |
| 6 | $1.2127 \pm 0.7422$ | $0.0353 \pm 0.0198$ | $1.2553 \pm 0.6716$ | $0.0368 \pm 0.0178$ |

is chosen uniformly from the interval $[10^0, 10^3]$ on a logarithmic scale and $\varepsilon$ is chosen from $[0.01, 0.1, 0.5, 1{:}0.5{:}10]$, a Matlab notation.

Since the forward stepwise feature selection method can only select 5 features sometimes when the significance level is set to 0.05, for a fair comparison, we set the numbers of selected features to be 5 and 6 for two batch of experiments. We then calculate the $MSE$ and $Q^2$ values of the SVRs trained in these selected features and report the results in Table 3. It can be observed that the regression results by NMMKLR are significantly better than those selected by SVR-LW, LASSO-LW, and the forward stepwise feature selection method in both cases.

**Table 3.** The results of two performance measures ($MSE$ and $Q^2$) on the house dataset when varying the number of selected features by NMMKLR and stepwise feature selection, SVR-LW, and LASSO-LW method. The best results on feature selection are highlighted (achieved by the paired $t$-test with 95% confidence level).

| #SF | NMMKLR | | Stepwise | |
|---|---|---|---|---|
| | $MSE$ | $Q^2$ | $MSE$ | $Q^2$ |
| 5 | $\mathbf{25.65} \pm 2.36$ | $\mathbf{0.3208} \pm 0.0329$ | $26.24 \pm 2.41$ | $0.3281 \pm 0.0326$ |
| 6 | $\mathbf{25.07} \pm 2.50$ | $\mathbf{0.3131} \pm 0.0290$ | $25.39 \pm 2.69$ | $0.3174 \pm 0.0344$ |
| #SF | SVR-LW | | LASSO-LW | |
| | $MSE$ | $Q^2$ | $MSE$ | $Q^2$ |
| 5 | $26.95 \pm 3.12$ | $0.3365 \pm 0.0368$ | $26.25 \pm 2.57$ | $0.3283 \pm 0.0345$ |
| 6 | $26.75 \pm 2.94$ | $0.3342 \pm 0.0360$ | $25.83 \pm 2.41$ | $0.3232 \pm 0.0344$ |

## 4 Conclusion

This paper presents a new framework of non-monotonic feature selection for regression models. By fixing the number of selected features and adopting the SVR, we develop an efficient non-monotonic feature selection algorithm for SVR via the multiple kernel learning framework to approximately solve the original combinatorial optimization problem. We further propose a strategy to derive a discrete solution for the relaxed problem with performance guarantee. The empirical study on both synthetic and real-world datasets shows the effectiveness of the proposed algorithm.

**Acknowledgement.** The work described in this paper was fully supported by the National Grand Fundamental Research 973 Program of China (No. 2014CB340401 and No. 2014CB340405), the Basic Research Program of Shenzhen (No. JCYJ20120619152419087), a Project 985 grant from University of Electronic Science and Technology (No. A1098531023601041), the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK 413212 and CUHK 415113), and Microsoft Research Asia Regional Seed Fund in Big Data Research (Grant No. FY13-RES-SPONSOR-036).

## References

1. Bach, F.R., Lanckriet, G.R.G., Jordan, M.I.: Multiple kernel learning, conic duality, and the SMO algorithm. In: ICML, pp. 41–48. ACM, New York (2004)
2. Bi, J., Bennett, K.P., Embrechts, M.J., Breneman, C.M., Song, M.: Dimensionality reduction via sparse support vector machines. J. Mach. Learn. Res. 3, 1229–1243 (2003)
3. Draper, N., Smith, H.: Applied Regression Analysis, 3rd edn. Wiley Interscience (1998)
4. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)
5. Harrison, D.J., Rubinfeld, D.L.: Hedonic housing prices and the demand for clean air. Journal of Environmental Economics and Management 5(1), 81–102 (1978)
6. Lanckriet, G.R.G., Cristianini, N., Bartlett, P., Ghaoui, L.E., Jordan, M.I.: Learning the kernel matrix with semidefinite programming. J. Mach. Learn. Res. 5, 27–72 (2004)
7. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Statistics and Computing 14(3), 199–222 (2004)
8. Tan, M., Wang, L., Tsang, I.W.: Learning sparse svm for feature selection on very high dimensional datasets. In: ICML, pp. 1047–1054 (2010)
9. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. Roy. Statist. Soc. Ser. B 58(1), 267–288 (1996)
10. Wolf, L., Shashua, A.: Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach. J. Mach. Learn. Res. 6, 1855–1887 (2005)
11. Xu, Z., Jin, R., King, I., Lyu, M.: An extended level method for efficient multiple kernel learning. In: NIPS, pp. 1825–1832 (2009)
12. Xu, Z., Jin, R., Ye, J., Lyu, M.R., King, I.: Non-monotonic feature selection. In: ICML, pp. 1145–1152 (2009)
13. Xu, Z., Jin, R., Zhu, S., Lyu, M.R., King, I.: Smooth optimization for effective multiple kernel learning. In: AAAI (2010)
14. Xu, Z., King, I., Lyu, M.R., Jin, R.: Discriminative semi-supervised feature selection via manifold regularization. IEEE Transactions on Neural Networks 21(7), 1033–1047 (2010)
15. Yang, H., Chan, L., King, I.: Support vector machine regression for volatile stock market prediction. In: Yin, H., Allinson, N.M., Freeman, R., Keane, J.A., Hubbard, S. (eds.) IDEAL 2002. LNCS, vol. 2412, pp. 391–396. Springer, Heidelberg (2002)
16. Yang, H., Huang, K., King, I., Lyu, M.R.: Localized support vector regression for time series prediction. Neurocomputing 72, 2659–2669 (2009)
17. Yang, H., Xu, Z., Ye, J., King, I., Lyu, M.R.: Efficient sparse generalized multiple kernel learning. IEEE Transactions on Neural Networks 22(3), 433–446 (2011)