Neural Networks 71 (2015) 214–224

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Budget constrained non-monotonic feature selection

Haiqin Yang [a,b,*], Zenglin Xu [c,d,**], Michael R. Lyu [a,b], Irwin King [a,b]

[a] Shenzhen Key Laboratory of Rich Media Big Data Analytics and Application, Shenzhen Research Institute, The Chinese University of Hong Kong,

[b] Computer Science & Engineering, The Chinese University of Hong Kong, Hong Kong

[c] Big Data Research Center, University of Electronic Science & Technology, Chengdu, Sichuan, China

[d] School of Computer Science and Engineering, University of Electronic Science & Technology, Chengdu, Sichuan, China

ARTICLE INFO

Article history:
Available online 4 September 2015

Keywords:
Feature selection
Multiple kernel learning
Budget constraint
Non-monotonic

ABSTRACT

Feature selection is an important problem in machine learning and data mining. We consider the problem of selecting features under the budget constraint on the feature subset size. Traditional feature selection methods suffer from the "*monotonic*" property. That is, if a feature is selected when the number of specified features is set, it will always be chosen when the number of specified feature is larger than the previous setting. This sacrifices the effectiveness of the non-monotonic feature selection methods. Hence, in this paper, we develop an algorithm for non-monotonic feature selection that approximates the related combinatorial optimization problem by a Multiple Kernel Learning (MKL) problem. We justify the performance guarantee for the derived solution when compared to the global optimal solution for the related combinatorial optimization problem. Finally, we conduct a series of empirical evaluation on both synthetic and real-world benchmark datasets for the classification and regression tasks to demonstrate the promising performance of the proposed framework compared with the baseline feature selection approaches.

1. Introduction

Feature selection is an important task in machine learning and data mining since one is often restricted with budgeted computational resources, such as the memory size, the CPU speed, the communication rate, etc., in a large number of real-world applications. The goal of feature selection is to choose from the input data a subset of informative features (Huang, Yang, King, & Lyu, 2008; Yang, King, & Lyu, 2011). It is often used to reduce the computational cost or save storage space for problems with high dimensional data for problems with either high dimensionality or limited computational power. This is helpful to prevent overfitting for high-dimensional data with relatively small training samples (Tibshirani, 1996; Yang, Lyu, & King, 2013; Yang, Xu, King, & Lyu, 2010). 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 group methods of data handling (Ivakhnenko, 1995; Reddy & Ravi, 2012; Tan, Tsang, & Wang, 2014; Thi, Vo, & Dinh, 2014; Wang, Bensmail, & Gao, 2014; Wang, Zhao, Hoi, & Jin, 2014; Wolf & Shashua, 2005). Comprehensive survey papers of feature selection can be found in Blum and Langley (1997), Guyon and Elisseeff (2003) and Kohavi and John (1997). The procedure of feature selection is analogous to pruning approach in neural networks, which aims to trim a network within the assumed initial architecture Augasta and Kathirvalavakumar (2013). Moreover, it is important to note that feature selection is different from feature extraction (He & Niyogi, 2003; Jolliffe, 1986; Kohonen, 2006), which maps the input data into a reduced representation set of features. Comparing with feature extraction, feature selection keeps the same space as the input data and thus has better interpretability for some specific applications.

In this paper, we consider the problem of feature selection under the budget constraint on the feature subset size. This setting is important for two reasons. On the one hand, budgeted learning is a new research aspect of machine learning since people are often facing a fixed budget in the presence of non-uniform cost functions for the acquisition of feature values, labels, or entire instances, and

* Correspondence to: Shenzhen Key Laboratory of Rich Media Big Data Analytics and Applications, Shenzhen Research Institute; Department of Computer Science and Engineering, The Chinese University of Hong Kong. Tel.: +852 31634251.
** Corresponding author at: Big Data Research Center, University of Electronic Science & Technology, Chengdu, Sichuan, China.
E-mail addresses: hqyang@ieee.org (H. Yang), zlxu@uestc.edu.cn (Z. Xu).

http://dx.doi.org/10.1016/j.neunet.2015.08.004
0893-6080/

for prediction errors (Dekel & Singer, 2006; Margineantu, Greiner, Singliar, & Melville, 2010). On the other hand, the number of required features also depends on the objective of the task, and there is no single number of features that are optimal for all tasks. For example, for data visualization, only two or three features are necessary. In this work, we assume that an external oracle decides the number of selected features.

Given the budget of the feature subset size, denoted by m, the goal of feature selection 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:

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

A number of performance criteria have been proposed for feature selection, including mutual information (Koller & Sahami, 1996), maximum margin (Guyon, Weston, Barnhill, & Vapnik, 2002; Weston et al., 2000), kernel alignment (Cristianini, Shawe-Taylor, Elisseeff, & Kandola, 2001; Neumann, Schnörr, & Steidl, 2005), worst case classification bounds (Bhattacharyya, 2004; Xu, King, & Lyu, 2007), graph-spectrum based measures (Zhao & Liu, 2007), Parzen window (Yu, Ding, & Loscalzo, 2008), clustering-based measures (Boutsidis, Mahoney, & Drineas, 2009; Fisher, 1996), PCA-based measures (Malhi & Gao, 2004), and the Hilbert Schmidt independence criterion (Song, Smola, Gretton, Borgwardt, & Bedo, 2007), etc. Among them, due to the effectiveness, the maximum-margin-based criterion is probably one of the most widely used criteria for feature selection.

The computational challenge in solving the optimization problem in Eq. (1) arises from its combinatorial nature, i.e., a binary selection of features that maximizes the performance criterion $\mathcal{Q}$ given the number of required features. A number of feature selection algorithms have been proposed to approximately solve Eq. (1). Most of them first compute a score or weight for each feature, and then select the features with the largest scores. For instance, a common approach is to first learn an SVM model, and select m features with the largest absolute weights. This idea was justified in Vapnik (1998) by sensitivity analysis and was also utilized for feature selection. A similar idea was used in SVM-Recursive Feature Elimination (SVM-RFE) (Guyon et al., 2002), where features with smallest weights were removed iteratively. In Fung and Mangasarian (2000) and Ng (2004), regularization on the L_1-norm of weights was suggested to replace the L_2-norm for feature selection when learning an SVM model. Another feature selection model related to the L_1-norm is lasso (Tibshirani, 1996), which selects features by constraining the L_1-norm of weights. By varying the L_1-norm of weights, a regularization path of selected features can be tracked. A similar model is LARS (Efron, Hastie, Johnstone, & Tibshirani, 2004), which can be regarded as unconstrained version of lasso. Other models related to the L_1-norm regularization include the direct optimization over the L_1-norm of the feature indicator (Sonnenburg, Rätsch, Schäfer, & Schölkopf, 2006; Xu, King, Lyu, & Jin, 2010). In addition to the optimization on the L_2-norm and the L_1-norm, several studies (Bradley & Mangasarian, 1998; Chan, Vasconcelos, & Lanckriet, 2007; Huang, King, & Lyu, 2008; Neumann et al., 2005; Weston, Elisseeff, Schölkopf, & Tipping, 2003) explored the L_0-norm when computing the weights of features. In Bradley and Mangasarian (1998), the authors proposed Feature Selection Concave method (FSV) that uses an approximate of the L_0-norm of the weights. It was improved in Neumann et al. (2005) and Weston et al. (2003) via an additional regularizer or a different approximation of the L_0-norm. In addition to selecting features by weights, in Rakotomamonjy (2003), Vapnik (1998) and Weston et al. (2000), the authors proposed to select features based on $R^2\|\mathbf{w}\|^2$, where R is the radius of the smallest sphere that contains all the data points.

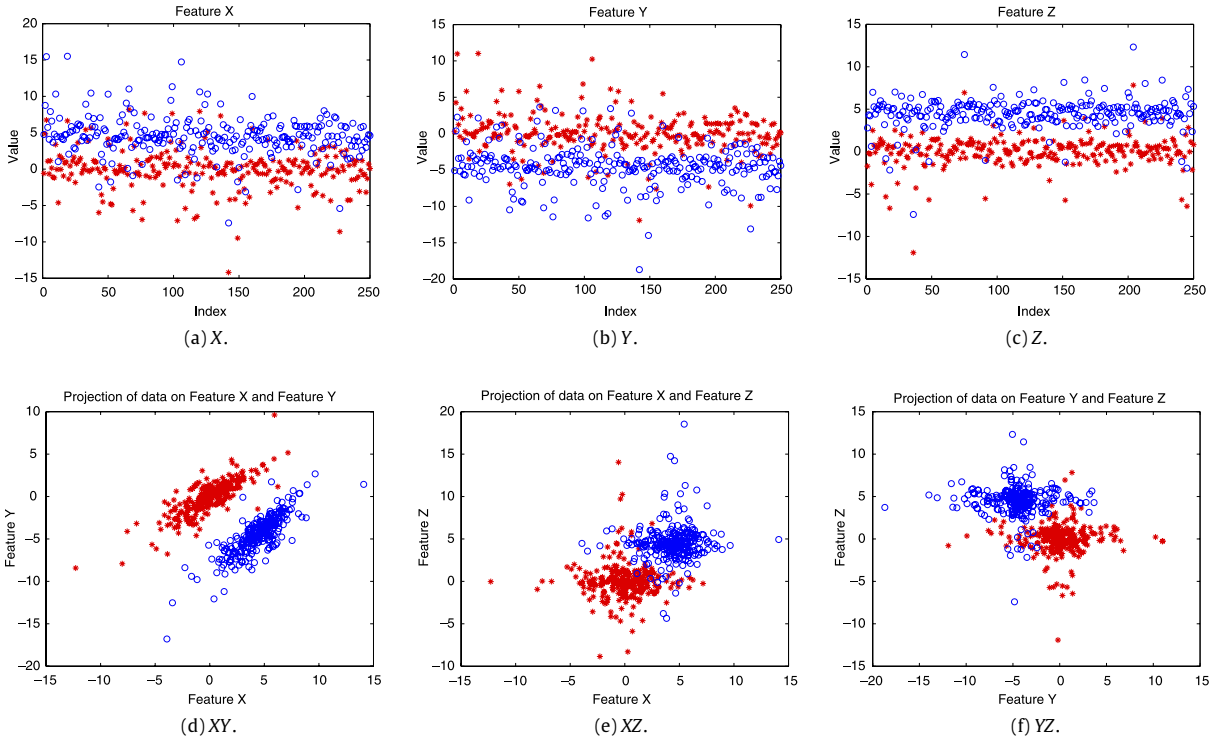
Although the above approximate approaches have been successfully applied to a number of applications of feature selection, they are limited by the **monotonic** property of feature selection that is defined below:

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.

To see the monotonic property of most existing algorithms for feature selection, first note that these algorithms rank features according to their weights/scores that are computed independently from the number of selected features m. Hence, if a feature f is chosen when the number of selected features is k, it will also be chosen if the number of selected features m is larger than k. In other words, $f \in \mathcal{S}_k \rightarrow f \in \mathcal{S}_m$ if $k < m$, and therefore $\mathcal{S}_k \subseteq \mathcal{S}_m$. As argued in Guyon and Elisseeff (2003), since variables that are less informative by themselves can be informative together, a monotonic feature selection algorithm may be suboptimal in identifying the set of most informative features. To further motivate the need of non-monotonic feature selection, we consider a binary classification problem with three features X, Y, Z. Fig. 1(a)–(c) show the projection of data points on individual features X, Y and Z, respectively. We clearly see that Z is the most informative feature to the two classes. Fig. 1(d)–(f) show the projection of data distribution on the plane of two joint features XY, XZ, and YZ, respectively. We observe that XY are the two most informative features. Note that although Z is the single most informative feature, its combinations with any other feature are not as informative as XY, which justifies the need of non-monotonic feature selection.

In this paper, we propose a **non-monotonic** feature selection method that solves the optimization problem in Eq. (1) approximately. In particular, we alleviate the monotonic property by computing scores for individual features that depend on the number of selected features m. We first convert the combinatorial optimization problem in Eq. (1) into a formulation that is closely related to multiple kernel learning (MKL) (Lanckriet, Cristianini, Bartlett, Ghaoui, & Jordan, 2004; Sonnenburg et al., 2006; Xu, Jin, Ye, Lyu, & King, 2009; Yang, Xu, King, & Lyu, 2014; Yang, Xu, Ye, King, & Lyu, 2011). The key idea is to first construct a separate kernel matrix for each feature, and then find the binary combination of kernels that minimizes the margin classification error. We relax the original combinatorial optimization problem into a convex optimization problem that can be solved efficiently by expressing it as a Quadratically Constrained Quadratic Programming (QCQP) problem. We present a strategy that selects a subset of features based on the solution of the relaxed problem, which can still maintain the non-monotonic property. This is different from the recent work in Tan et al. (2014). We furthermore 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 baseline methods for feature selection. Finally, we would like to clarify that although our work involves the employment of MKL, the focus of our work is not to develop a new algorithm for MKL, but an efficient algorithm for non-monotonic feature selection.

The rest of this paper is organized as follows. We present the non-monotonic feature selection for classification and regression in Sections 2 and 3, respectively. Sections 4 and 5 present experimental results with a number of benchmark datasets for classification and regression, respectively. We conclude our work in Section 6.



(a) X. (b) Y. (c) Z.

(d) XY. (e) XZ. (f) YZ.

Fig. 1. A failed example for monotonic feature selection. (a)–(c) show the projection of data distribution on individual features X, Y, and Z, respectively. (d)–(f) show the projection on the plane of two joint features, respectively. The two classes are denoted by symbols ∘ and ∗, respectively.

2. Non-monotonic feature selection via multiple kernel learning

In this section, we first show that multiple kernel learning framework can be utilized for non-monotonic feature selection. We then present an efficient algorithm to approximately solve the related discrete optimization problem. Finally, we prove the performance guarantee of the approximate solution for the discrete optimization problem.

Let N denote the number of training examples. We denote by $\mathbf{x}_i \in \mathbb{R}^N$ the vector of the ith attributes for all the training examples. Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_d)^\top$, where d is the total number of features. We denote $\mathbf{1}_d \in \mathbf{R}^d$ as a d-dimensional vector with all elements being one. We also omit the suffix when the dimensionality d of $\mathbf{1}_d$ can be easily inferred from the context. For a linear kernel, the kernel matrix (Gram 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. To select a subset of $m < d$ features, we modify $\mathbf{K}$ as:

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

where $p_i \in \{0, 1\}$ is a binary variable that indicates if the ith feature is selected, and $\mathbf{p} = (p_1, \ldots, p_d)$. As revealed in Eq. (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 for classification, given a kernel matrix $\mathbf{K}(\mathbf{p}) = \sum_{i=1}^d p_i\mathbf{K}_i$, the classification model is found by solving the following optimization problem:

$$\max_{\alpha} \; 2\alpha^\top\mathbf{1} - (\alpha \circ \mathbf{y})^\top (\mathbf{K}(\mathbf{p}) + \tau\mathbf{I})(\alpha \circ \mathbf{y}) \tag{3}$$
$$\text{s.t. } \alpha^\top\mathbf{y} = 0, \quad 0 \le \alpha \le C,$$

where $\mathbf{I}$ is the identity matrix; α is the dual variable for the margin error; both C and τ are manually set constants; ∘ stands for the element-wise product between two vectors. Notation $0 \le \alpha \le C$ is a shorthand for $0 \le \alpha_i \le C,\ i = 1, \ldots, N$. If $\mathbf{p} = \mathbf{1}$, then Eq. (3) reduces to a standard SVM.

We denote by $\omega(\mathbf{p})$ the value of the objective function in Eq. (3), which represents the overall margin errors of the classification model. The subset of m most informative features are chosen by minimizing $\omega(\mathbf{p})$, i.e.,

$$\min_{\mathbf{p}\in\{0,1\}^d} \; \omega(\mathbf{p}) \quad \text{s.t. } \mathbf{p}^\top\mathbf{1} = m. \tag{4}$$

Evidently, the challenge with solving the above problem is the constraint $\mathbf{p} \in \{0, 1\}^d$. We thus relax p_i in Eq. (4) into a continuous variable, and have the following continuous optimization problem:

$$\min_{0\le\mathbf{p}\le 1} \; \omega(\mathbf{p}) \quad \text{s.t. } \mathbf{p}^\top\mathbf{1} = m. \tag{5}$$

Remark. It is important to note that although the objective function in Eq. (3) appears to be a linear function in $\mathbf{p}$, $\omega(\mathbf{p})$ is NOT a linear function of $\mathbf{p}$ because of the maximization. As a result, Eq. (5) may have a non-discrete solution. To see this, consider the following problem

$$\min_{0\le\mathbf{p}\le 1, \mathbf{p}^\top\mathbf{1}=1} \; \max_{\mathbf{z}\in\mathbb{R}^d} \; 2\mathbf{p}^\top\mathbf{z} - \|\mathbf{z}\|_2^2. \tag{6}$$

Since $\max_{\mathbf{z}} 2\mathbf{p}^\top\mathbf{z} - \|\mathbf{z}\|_2^2 = \|\mathbf{p}\|_2^2$, the optimal solution to Eq. (6) is $p_i = 1/d$, which is definitely not discrete.

Below, we will discuss how to solve the relaxed min–max problem in Eq. (5) efficiently, followed by the algorithm that derives a discrete solution for Eq. (4) based on the optimal solution to Eq. (5).

It can be shown that Eq. (5) is equivalent to the following problem according to Lanckriet et al. (2004):

$$\min_{\mathbf{p},t,\nu,\delta,\theta} \quad t + 2C\delta^\top \mathbf{1} \tag{7}$$

$$\text{s.t.} \quad \begin{pmatrix} \mathbf{K}(\mathbf{p}) \circ (\mathbf{y}\mathbf{y}^\top) + \tau \mathbf{I} & \mathbf{1} + \nu - \delta + \theta \mathbf{y} \\ (\mathbf{1} + \nu - \delta + \theta \mathbf{y})^\top & t \end{pmatrix} \succeq 0,$$

$$\nu \geq 0,\ \delta \geq 0, \quad \mathbf{p}^\top \mathbf{1} = m, \quad 0 \leq \mathbf{p} \leq 1.$$

However, the above formulation is a semi-definite programming (SDP) problem and is therefore expensive to solve. The following theorem shows that Eq. (7) can be reformulated into a Quadratically Constrained Quadratic Programming (QCQP) problem, which is also justified in Bach, Lanckriet, and Jordan (2004).

Theorem 1. *The dual problem of Eq. (7) is*

$$\max_{\alpha,\lambda,\gamma} \quad 2\alpha^\top \mathbf{1} - \tau \alpha^\top \alpha - m\lambda - \gamma^\top \mathbf{1} \tag{8}$$

$$\begin{aligned} \text{s.t.} \quad & \alpha^\top \mathbf{y} = 0, \quad 0 \leq \alpha \leq C, \\ & (\alpha \circ \mathbf{y})^\top \mathbf{K}_i (\alpha \circ \mathbf{y}) \leq \lambda + \gamma_i, \quad i = 1, \ldots, d, \\ & \gamma_i \geq 0, \quad i = 1, \ldots, d. \end{aligned}$$

The KKT conditions are

$$\begin{aligned} & (\mathbf{K}(\mathbf{p}) \circ \mathbf{y}\mathbf{y}^\top + \tau \mathbf{I})\alpha = \mathbf{1} + \nu - \delta + \theta \mathbf{y}, \\ & t = \alpha^\top (\mathbf{1} + \nu - \delta + \theta \mathbf{y}), \\ & \nu \circ \alpha = 0, \qquad \alpha \circ \delta = C\delta, \qquad \gamma \circ (\mathbf{1} - \mathbf{p}) = 0, \\ & p_i(\lambda + \gamma_i - (\alpha \circ \mathbf{y})^\top \mathbf{K}_i (\alpha \circ \mathbf{y})) = 0, \quad i = 1, \ldots, d. \end{aligned}$$

The proof can be found in Appendix A.

We can now derive properties of the primal and dual solutions using the KKT conditions in Theorem 1. Before we state the results, we first rank the features in the descending order of

$$\tau_i = \alpha^\top (\mathbf{K}_i \circ (\mathbf{y}\mathbf{y}^\top))\alpha. \tag{9}$$

We denote by $i_1, \ldots, i_d$ 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:

$$\mathcal{A} = \{i_k | 1 \leq k < k_{\min}\}, \tag{10}$$

$$\mathcal{B} = \{i_k | k_{\min} \leq k \leq k_{\max}\}, \tag{11}$$

$$\mathcal{C} = \{i_k | k_{\max} < k \leq d\}. \tag{12}$$

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

$$\lambda \in [\tau_{1+k_{\max}}, \tau_m], \qquad p_i = \begin{cases} 1, & i \in \mathcal{A}, \\ 0, & i \in \mathcal{C}. \end{cases} \tag{13}$$

The proof can be found in Appendix B.

The following corollary shows the relationship between Eq. (8) and the dual problem of SVM in Eq. (3).

Corollary 2. *When $m = d$, i.e., when all the features are selected, Eq. (8) is reduced to the dual problem of a linear SVM in Eq. (3).*

Proof. First, we combine these two constraints $\lambda + \gamma_i \geq \alpha^\top (\mathbf{K}_i \circ (\mathbf{y}\mathbf{y}^\top))\alpha$ and $\gamma_i \geq 0$, and express γ_i as $\gamma_i = \max(0, \tau_i - \lambda)$. We then rewrite Eq. (8) as follows:

$$\max_{\alpha,\lambda,\gamma} \quad 2\alpha^\top \mathbf{1} - \tau \alpha^\top \alpha + \lambda(d - m) - \sum_{i=1}^{d} \max(\lambda, \tau_i) \tag{14}$$

$$\text{s.t.}\ \alpha^\top \mathbf{y} = 0, \quad 0 \leq \alpha \leq C, \qquad \lambda \geq 0, \qquad \gamma \geq 0.$$

When $m = d$, we have $\lambda = 0$ since the linear term $\lambda(m - d) = 0$, and $\max(\lambda, \tau_i) = \tau_i$ since $\tau_i \geq 0$. Substituting $\lambda = 0$ and $\max(\lambda, \tau_i) = \tau_i$ in Eq. (14), we have the formulation of a linear SVM in Eq. (3). ■

Remark. The desired number of selected features, i.e., m, controls the sparseness of features. It is related to the ν-SVM (Hsuen Chen, Lin, & Schölkopf, 2005), which bounds the ratio of support vectors.

The following theorem shows how to derive $\mathbf{p}$ from the solution of the dual problem in Eq. (7).

Theorem 2. *Given the solution to the dual problem in Eq. (8), denoted by α, γ, and λ, the solution to the primal problem in Eq. (7) can be found by solving the following linear programming problem:*

$$\min_{\mathbf{p},\nu,\delta} \quad \alpha^\top (\mathbf{K}(\mathbf{p}) \circ \mathbf{y}\mathbf{y}^\top + \tau \mathbf{I})\alpha + 2C\mathbf{1}^\top \delta \tag{15}$$

$$\begin{aligned} \text{s.t.} \quad & (\mathbf{K}(\mathbf{p}) \circ \mathbf{y}\mathbf{y}^\top + \tau \mathbf{I})\alpha = \mathbf{1} + \nu - \delta + \theta \mathbf{y}, \\ & \nu \circ \alpha = 0, \quad \alpha \circ \delta = C\delta, \quad \delta \geq 0, \quad \nu \geq 0, \\ & 0 \leq \mathbf{p} \leq 1, \quad \mathbf{1}^\top \mathbf{p} = m, \quad \gamma \circ (\mathbf{1} - \mathbf{p}) = 0, \\ & p_i(\lambda + \gamma_i - (\alpha \circ \mathbf{y})^\top \mathbf{K}_i (\alpha \circ \mathbf{y})) = 0, \quad i = 1, \ldots, d. \end{aligned}$$

Proof. The problem in Eq. (15) can be verified directly using the KKT conditions in Theorem 1. ■

Although Eq. (15) is a linear programming problem, the solution for $\mathbf{p}$ may be not completely discrete due to the constraint

$$(\mathbf{K}(\mathbf{p}) \circ \mathbf{y}\mathbf{y}^\top + \tau \mathbf{I})\alpha = \mathbf{1} + \nu - \delta + \theta \mathbf{y}. \tag{16}$$

The following theorem shows the optimal solution to Eq. (15) is discrete if constraint Eq. (16) is dropped.

Theorem 3. *Consider the following problem:*

$$\min_{\mathbf{p},\nu,\delta} \quad \alpha^\top (\mathbf{K}(\mathbf{p}) \circ \mathbf{y}\mathbf{y}^\top + \tau \mathbf{I})\alpha + 2C\mathbf{1}^\top \delta \tag{17}$$

$$\begin{aligned} \text{s.t.} \quad & \nu \circ \alpha = 0, \qquad \alpha \circ \delta = C\delta, \quad \delta \geq 0, \ \nu \geq 0, \\ & 0 \leq \mathbf{p} \leq 1, \quad \mathbf{1}^\top \mathbf{p} = m, \quad \gamma \circ (\mathbf{1} - \mathbf{p}) = 0, \\ & p_i(\lambda + \gamma_i - (\alpha \circ \mathbf{y})^\top \mathbf{K}_i (\alpha \circ \mathbf{y})) = 0, \quad i = 1, \ldots, d, \end{aligned}$$

where λ, γ, and α are the optimal solution to Eq. (8). An optimal solution $\mathbf{p}$ to Eq. (17) can be obtained by selecting the first m features with the largest τ_i (defined in Eq. (9)) and assigning $p_i = 1$ for the selected features.

Proof. First, notice that an optimal solution for δ and ν to Eq. (17) is $\delta = \nu = 0$. Since Eq. (13) gives binary solutions for p_i if $i \in \mathcal{A} \cup \mathcal{C}$, the only remaining undecided variables for Eq. (17) are $\{p_i | i \in \mathcal{B}\}$. Second, notice that the objective function in Eq. (17) remains the same no matter which subset of $s = m + 1 - k_{\min}$ features are selected from $\mathcal{B}$. This is because $\tau_j = \alpha^\top (\mathbf{K}_j \circ \mathbf{y}\mathbf{y}^\top)\alpha = \lambda$ for any $j \in \mathcal{B}$. This implies the selection of m features with the largest τ_i provides an optimal solution to Eq. (17). ■

The above theorem suggests a simple algorithm of deriving a discrete solution for $\mathbf{p}$ based on the value of $\alpha^\top (\mathbf{K}_i \circ (\mathbf{y}\mathbf{y}^\top))\alpha$, which is summarized in Algorithm 1.

Algorithm 1 Non-monotonic feature selection via MKL

Input:

- $X \in \mathbb{R}^{d \times N}$, $\mathbf{y} \in \{-1, +1\}^N$: training data
- m: the number of selected features

Algorithm:

- Solve α for (8)
- Compute $\tau_i = (\sum_{j=1}^{N} X_{i,j} \alpha_j y_j)^2$
- Select the first m features with the largest τ_i.

Remark. We can rewrite τ_i as follows $\tau_i = \alpha^\top(\mathbf{K}_i \circ \mathbf{y}\mathbf{y}^\top)\alpha = (\sum_{j=1}^N \alpha_j y_j X_{i,j})^2 = w_i^2$, where w_i is the weight computed for the ith feature. Hence, the algorithm described in Algorithm 1 essentially selects the features with the largest absolute weights. Compared with the simple greedy algorithm that selects features with the largest absolute weights computed by SVM, the key difference is that α used in our algorithm is computed by Eq. (8), not by Eq. (3).

The following theorem shows the performance guarantee of the discrete solution constructed by Algorithm 1 for the combinatorial optimization problem in Eq. (4).

Theorem 4. *The discrete solution constructed by Algorithm* 1, *denoted by* $\mathbf{p}$, *has the following performance guarantee for the combinatorial optimization problem in Eq.* (4)*:*

$$\frac{\omega(\mathbf{p})}{\omega(\tilde{\mathbf{p}}^*)} \leq \frac{1}{1-\sigma_{\max}(\mathbf{M}^{-1/2}\mathbf{B}\mathbf{M}^{-1/2})},$$

where

$$\mathbf{M} = \mathbf{K}(\mathbf{p}^*) \circ (\mathbf{y}\mathbf{y}^\top) + \tau\mathbf{I}, \quad \mathbf{B} = \sum_{j\in\mathcal{B}} p_j^* \mathbf{K}_j.$$

The operator $\sigma_{\max}(\cdot)$ *calculates the largest eigenvalue.* $\mathbf{p}^*$ *is the optimal solution to the relaxed optimization problem in Eq.* (5)*, and* $\tilde{\mathbf{p}}^*$ *is the global optimal solution to the combinatorial optimization problem in Eq.* (4)*.*

The proof can be found in Appendix C.

As indicated by Theorem 4, the bound for the suboptimality of the approximate solution depends on the number of selected features through the set $\mathcal{B}$. Thus, by incorporating the required number of selected features, the resulting approximate solution could be more accurate than without it. This suggests that the proposed algorithm produces a better approximation to the underlying combinatorial optimization problem Eq. (4).

3. Non-monotonic feature selection for regression

In this section, we discuss how to extend our proposed feature selection method to solve the regression task. Here, we adopt the performance measurement related to support vector regression (SVR) with ε-insensitive loss function (Smola & Schölkopf, 2004; Vapnik, 1999; Yang, Chan, & King, 2002; Yang, Huang, King, & Lyu, 2009) as our objective function for feature selection.

Similar to the optimization problem in Eq. (5) for classification, we define the optimization related to regression as follows:

$$\min_{0\leq\mathbf{p}\leq 1} \tilde{\omega}(\mathbf{p}) \quad \text{s.t. } \mathbf{p}^\top\mathbf{1} = m. \tag{18}$$

Note that in the above, we also employ continuous indicator to approximate the original combinatorial problem related to feature selection.

The optimization problem related to $\tilde{\omega}(\mathbf{p})$ is defined as:

$$\tilde{\omega}(\mathbf{p}) = \begin{cases} \max\limits_{\beta} & 2\mathbf{v}^\top\beta - \beta^\top\mathbf{Q}(\mathbf{p})\beta \\ \text{s.t.} & 0 \leq \beta \leq C, \quad \mathbf{u}^\top\beta = 0, \end{cases} \tag{19}$$

where the variable $\beta = [\alpha; \alpha^*] \in \mathbb{R}^{2N}$, and α, $\alpha^* \in \mathbb{R}^N$ are the corresponding Lagrange multipliers used to push and pull $f(\mathbf{x})$ towards the outcome of y, respectively. The linear coefficient $\mathbf{v}$ is defined as $[\mathbf{v}_1; \mathbf{v}_2]$, where $\mathbf{v}_1 = [-\varepsilon\mathbf{1} + \mathbf{y}]$ and $\mathbf{v}_2 = [-\varepsilon\mathbf{1} - \mathbf{y}]$. $\mathbf{u}$ in the equality constraint is defined as $[\mathbf{1}^\top, -\mathbf{1}^\top]$. The matrix $\mathbf{Q}(\mathbf{p}) \in \mathbb{R}^{2N\times 2N}$ is defined as

$$\mathbf{Q}(\mathbf{p}) = \begin{bmatrix} \mathbf{K}(\mathbf{p}) & -\mathbf{K}(\mathbf{p}) \\ -\mathbf{K}(\mathbf{p}) & \mathbf{K}(\mathbf{p}) \end{bmatrix}.$$

It is easy to observe that, given $\mathbf{p}$, the optimization problem in Eq. (19) is a quadratic programming problem with box constraints and an equality constraint, the same structure in Eq. (3). The only difference between Eqs. (3) and (19) is that the variables in Eq. (19) are in $2N$-dimensional, double length of the variables in Eq. (3). Therefore, based on Theorem 1, we have the following theorem:

Theorem 5. *The optimization problem in Eq.* (18) *can be reduced to the following optimization problem:*

$$\begin{aligned} \max_{\alpha,\alpha^*,\lambda,\gamma} \quad & 2(\mathbf{v}_1^\top\alpha + \mathbf{v}_2^\top\alpha^*) - m\lambda - \gamma^\top\mathbf{1} \\ \text{s.t.} \quad & \mathbf{1}^\top(\alpha - \alpha^*) = 0, \quad 0 \leq \alpha, \alpha^* \leq C, \\ & (\alpha - \alpha^*)^\top\mathbf{K}_i(\alpha - \alpha^*) \leq \lambda + \gamma_i, \\ & \gamma_i \geq 0. \end{aligned} \tag{20}$$

Proof. Similar to the derivative from Eqs. (5)–(7), we can obtain an SDP problem by calculating the dual of Eq. (18) over β as follows

$$\begin{aligned} \min_{\mathbf{p},t,\nu,\delta,\theta} \quad & t + 2C\delta^\top\mathbf{1} \\ \text{s.t.} \quad & \begin{pmatrix} \mathbf{Q}(\mathbf{p}) & \mathbf{v} + \nu - \delta + \theta\mathbf{u} \\ (\mathbf{v} + \nu - \delta + \theta\mathbf{u})^\top & t \end{pmatrix} \succeq 0, \\ & \nu \geq 0, \quad \delta \geq 0, \quad \mathbf{p}^\top\mathbf{1} = m, \quad 0 \leq \mathbf{p} \leq 1. \end{aligned} \tag{21}$$

We can then derive the dual problem of Eq. (21) over $\mathbf{p}$ as follows

$$\begin{aligned} \max_{\beta,\lambda,\gamma} \quad & 2\mathbf{v}^\top\beta - m\lambda - \gamma^\top\mathbf{1} \\ \text{s.t.} \quad & \mathbf{u}^\top\beta = 0, \quad 0 \leq \beta \leq C, \\ & \beta^\top\mathbf{Q}_i\beta \leq \lambda + \gamma_i, \\ & \gamma_i \geq 0, \end{aligned}$$

where $\mathbf{Q}_i$ is defined as $\mathbf{Q}_i = \begin{bmatrix} \mathbf{x}_i\mathbf{x}_i^\top & -\mathbf{x}_i\mathbf{x}_i^\top \\ \mathbf{x}_i\mathbf{x}_i^\top & \mathbf{x}_i\mathbf{x}_i^\top \end{bmatrix}$.

The derivation is similar to that in Appendix A, where only α is replaced by β. Restoring the expression of β into α and α^*, we can obtain the programming problem in Eq. (20). ■

Remark. The non-monotonic feature selection via MKL for the regression task is equivalent to solving a linear objective function with quadratic constraints, which is a special case of the QCQP. Finally, a discrete solution for $\mathbf{p}$ can be approximated by ranking the values of $\beta^\top\mathbf{Q}_i\beta$. The procedure is summarized in Algorithm 2.

Algorithm 2 Non-monotonic feature selection via MKL for Regression task

Input:
- $X \in \mathbb{R}^{d\times N}$, $\mathbf{y} \in \mathbb{R}^N$: training data
- m: the number of selected features

Procedure:
- Solve α, α^* in (20)
- Compute $\tau_i = (\sum_{j=1}^N X_{i,j}(\alpha_j - \alpha_j^*))^2$
- Select the first m features with the largest τ_i.

4. Experiment on feature selection for classification

We denote by **NMMKL** the proposed algorithm for non-monotonic feature selection. The greedy algorithm that selects the features with the largest absolute weights $|w_i|$ computed by SVM is used as our baseline method, and is referred to as **SVM-LW**. We also compare our algorithm to the following baseline approaches for feature selection in classification:

Table 1
The test accuracy (%) for the toy dataset. #SF stands for the number of selected features.

#SF	NMMKL	SVM-LW	L_0-appr	Fisher	R^2W^2	FSV	L_1-SVM
1	**93.9** ± 1.9	86.4 ± 3.2	85.7 ± 2.9	**93.9** ± 1.9	90.3 ± 4.4	86.3 ± 2.7	86.3 ± 3.3
2	**99.7** ± 0.5	**99.7** ± 0.5	**99.7** ± 0.5	94.7 ± 1.8	97.5 ± 2.8	99.4 ± 1.4	**99.7** ± 0.5

- **Fisher** (Bishop, 1996): calculating a Fisher/Correlation score for each feature.
- **FSV** (Bradley & Mangasarian, 1998): approximating the L_0-norm of $\mathbf{w}$ by a summation of exponential functions.
- R^2W^2 (Weston et al., 2000): adjusting weight $\mathbf{w}$ by computing gradient descents on a bound of the leave-one-out error.
- L_0-**appr** (Weston et al., 2003): approximating the L_0-norm by minimizing a logarithm function.
- L_1-**SVM** (Fung & Mangasarian, 2000): replacing the L_2-norm of $\mathbf{w}$ with the L_1-norm in SVM.

For all the methods, features with the largest scores are selected. For L_1-SVM, we use the implementation in Fung and Mangasarian (2000); for other baseline algorithms, we adopt the implementations in Spider.[1]

4.1. Experiment on a synthetic dataset

We first run our experiments over the synthetic dataset that is illustrated in Fig. 1. We randomly select 400 examples from the synthetic dataset as the training data and the remaining 100 examples are used as the test data. We repeat the experiment 30 times. To avoid any side effects caused by scales of different dimensions, we normalize each feature to be a Gaussian distribution with zero mean and unit standard deviation, based on the training data. The regularization parameter C in all SVM-based feature selection methods is chosen by a 5-fold cross validation. Parameter τ in our formulation is also tuned by a 5-fold cross validation. The number of required features is varied from 1 to 2. A linear SVM using the features selected by different algorithms is used as the classifier to compute the classification accuracy on the test data. We report the results averaged over 30 runs in Table 1. When selecting one feature, we observe that both the proposed NMMKL and Fisher could identify the most informative feature, i.e., $\mathcal{S}_1 = \{Z\}$, for the toy data. In contrast, the other five algorithms rank Z as the least informative feature, which leads to relatively low classification accuracy. When selecting two features, NMMKL and most of the comparison algorithms are able to identify the best feature subset $\mathcal{S}_2 = \{X, Y\}$. In contrast, Fisher fails to identify $\{X, Y\}$ as the subset of two most informative features. This is because according to the monotonic property of Fisher, $\mathcal{S}_2$ selected by Fisher must be a superset of $\mathcal{S}_1$, and as a result $Z \in \mathcal{S}_2$ for Fisher. In conclusion, NMMKL successfully identifies the best feature subsets in both cases. This shows the importance of non-monotonic feature selection, which requires the ranking procedure in feature selection to be dependent on the number of selected features.

4.2. Experiment on real-world datasets

The datasets well studied from previous literatures of feature selection (Guyon et al., 2002; Weston et al., 2003) are employed in our experiments. We select datasets from three different data repositories for our evaluation: (a) four binary datasets from the UCI repository,[2] namely Ionosphere, Sonar, Wdbc, and Wpbc; (b) three datasets from the Semi-supervised Learning book,[3] namely Digit1, Usps, and Bci; and (c) two microarray datasets,[4] namely Colon and Lymphoma. Table 2 lists the statistics of the datasets.

Table 2
Datasets used in the experiments.

Data	*dim*	*Num*	Data	*dim*	*Num*
Iono	34	351	Wdbc	30	569
Wpbc	33	198	Sonar	60	208
Bci	117	400	Digit1	241	1500
Usps	241	1500	Coil	241	1500
Colon	2000	62	Lym	4026	96

Note that the two microarray datasets are rather challenging compared to the other datasets since they contain a small number of data points but have very high dimensionality. Therefore, it is important to study the effect of feature selection when the number of features is very large while the number of instances is modest.

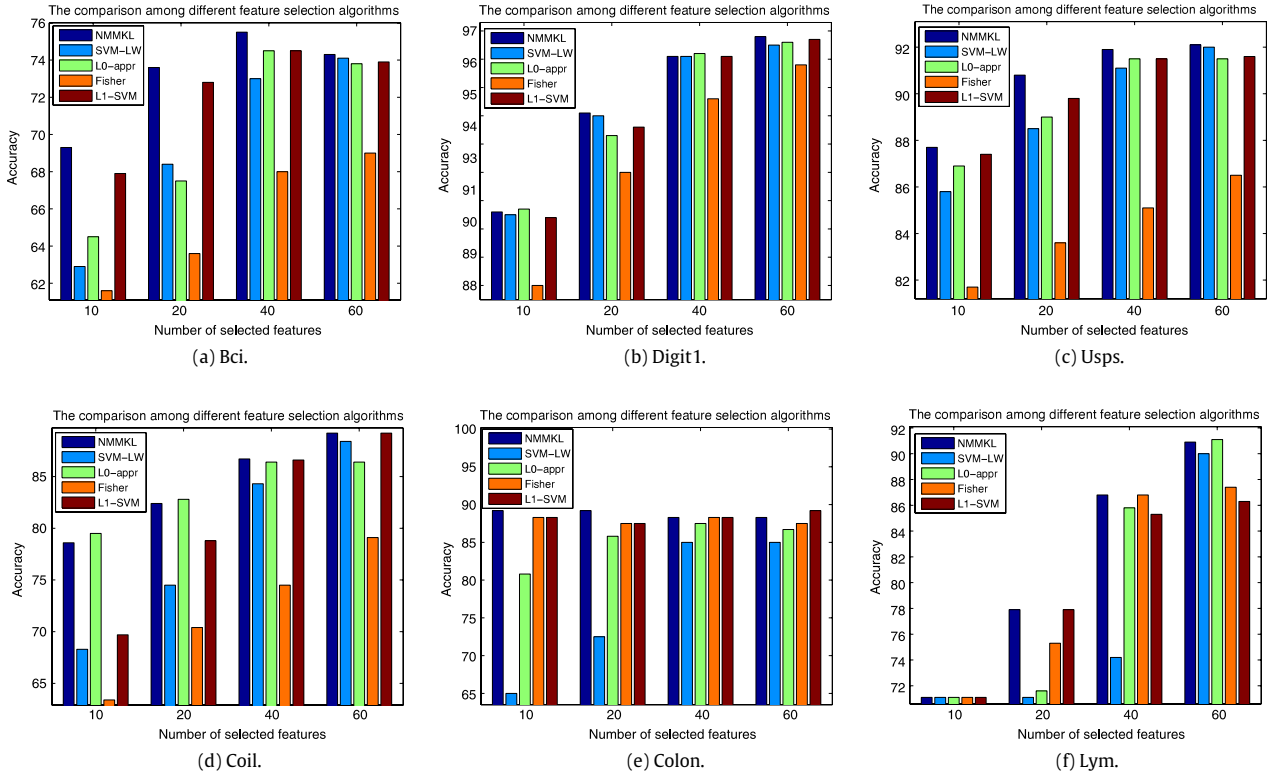
For all the datasets, 80% of the examples are randomly selected as the training data and the remainder are used as the test data. Every experiment is repeated with 30 random trials. The same procedure, which was applied to the synthetic dataset, is also applied to the nine real-world datasets to normalize data and decide parameters C and τ. To speed up the computation for the two microarray datasets (i.e., Colon and Lymphoma), Fisher is first used to select the 1000 features with the largest Fisher scores as the candidates for feature selection. Features selected by different algorithms are fed into a linear SVM for training, and the classification accuracy of test data is used as the evaluation metric. The number of selected features is set to be 10 and 20 for the four UCI datasets, and 10, 20, 40, and 60 for the other five datasets. This is because Bci, Digit1, Usps, and the two micro-array datasets contain examples with significantly higher dimensionality than the UCI datasets, and therefore allow for larger numbers of selected features.

We present the classification results for the four UCI datasets in Table 3 and the results of the remaining datasets in Fig. 2.[5] First, we compare the proposed feature selection method to SVM-LW. We observe that for almost all the cases, the proposed approach outperforms SVM-LW. For several datasets with different number of selected features (e.g., Colon and Sonar with 10 and 20 features), the improvement is significant. As revealed in Corollary 2, the proposed algorithm is similar to SVM-LW except that the weights α are computed differently. Thus, this result indicates that α computed by the proposed approach is more effective for feature selection than those computed by SVM. Second, we compare the proposed method to the other state-of-the-art approaches for feature selection. Among all the competitors, we found that methods L_0-appr and L_1-SVM overall deliver good performance across all the datasets. We find that overall the proposed approach performs slightly better than L_0-appr and L_1-SVM for most of the cases. For datasets Sonar and Bci, the improvement made by the proposed algorithm is statistically significant (Student's-t)

[1] www.kyb.tuebingen.mpg.de/bs/people/spider/.
[2] http://archive.ics.uci.edu/ml/.
[3] www.kyb.tuebingen.mpg.de/ssl-book/.
[4] www.kyb.tuebingen.mpg.de/bs/people/weston/l0/.
[5] Since R^2W^2 and *FSV* are time consuming on high dimensional datasets, we do not include their results.

Table 3
The classification accuracy (%) on real-world datasets. The best result and those not significantly worse than it (achieved by t-test with 95% confidence level), are highlighted by the bold font in each case.

Data	#SF	NMMKL	SVM-LW	L_0-appr	Fisher	R^2W^2	FSV	L_1-SVM
Sonar	10	**75.0** ± 2.3	71.4 ± 4.6	69.8 ± 5.9	69.3 ± 5.9	64.3 ± 7.1	71.4 ± 5.1	70.0 ± 6.0
	20	**75.0** ± 5.8	72.1 ± 5.8	74.1 ± 4.8	72.4 ± 3.4	70.7 ± 4.6	73.1 ± 4.2	72.1 ± 4.4
Iono	10	**86.1** ± 3.7	**85.3** ± 5.2	**85.6** ± 5.0	84.3 ± 5.0	**86.0** ± 4.3	82.4 ± 5.6	**86.6** ± 4.0
	20	**87.3** ± 4.1	**86.4** ± 4.7	85.7 ± 4.7	85.1 ± 3.8	85.1 ± 4.8	**86.7** ± 3.4	**86.6** ± 3.8
Wdbc	10	**97.0** ± 1.0	95.1 ± 0.8	96.0 ± 0.8	94.6 ± 1.7	93.5 ± 1.2	94.2 ± 1.0	96.3 ± 0.4
	20	**97.4** ± 0.6	**97.4** ± 0.5	**97.2** ± 1.0	**97.4** ± 0.6	94.6 ± 1.0	96.5 ± 1.0	**97.0** ± 0.5
Wpbc	10	**79.5** ± 4.8	78.3 ± 4.7	**79.8** ± 6.0	78.2 ± 5.2	78.0 ± 5.1	78.0 ± 5.1	**79.0** ± 6.4
	20	**81.2** ± 5.0	**80.7** ± 5.2	**80.4** ± 4.7	80.1 ± 4.7	79.3 ± 3.9	78.6 ± 4.6	78.0 ± 6.7



(a) Bci. (b) Digit1. (c) Usps.

(d) Coil. (e) Colon. (f) Lym.

Fig. 2. The classification accuracy of feature selection algorithms.

when compared to L_0-appr and L_1-SVM. Note that although the proposed algorithm does not always deliver the best performance, it consistently performs well across all the datasets for different numbers of selected features. In contrast, we observe that both L_0-appr and L_1-SVM could have poor performance for certain datasets. For instance, when the number of selected features is 10, L_0-appr does not perform well on Colon and Bci, and L_1-SVM fails to deliver good performance for Sonar. Finally, we conduct the pairwise paired t-test to compare the performance of the proposed algorithms to the five baselines. We found that the proposed non-monotonic feature selection algorithm is better or not significantly worse than other methods in almost all cases when p value is 0.05. We would like to note that the variance in classification accuracy is significantly larger for the two micro-array datasets than the others. This may be attributed by the very high dimensions of the two datasets.

5. Experiments on feature selection for regression

We denote by **NMMKLR** the proposed algorithm for non-monotonic feature selection on regression. We conduct empirical evaluation on both synthetic data and real-world benchmark datasets and compare NMMKLR with the following baseline methods:

- **Stepwise**: the forward stepwise feature selection method (Bi, Bennett, Embrechts, Breneman, & Song, 2003; Guyon & Elisseeff, 2003),[6]
- **SVR-LW**: features are selected with the largest absolute weights $|w_i|$ computed by SVR (Smola & Schölkopf, 2004),
- **LASSO-LW**: features are selected with the largest absolute weights $|w_i|$ computed by LASSO (Tibshirani, 1996).

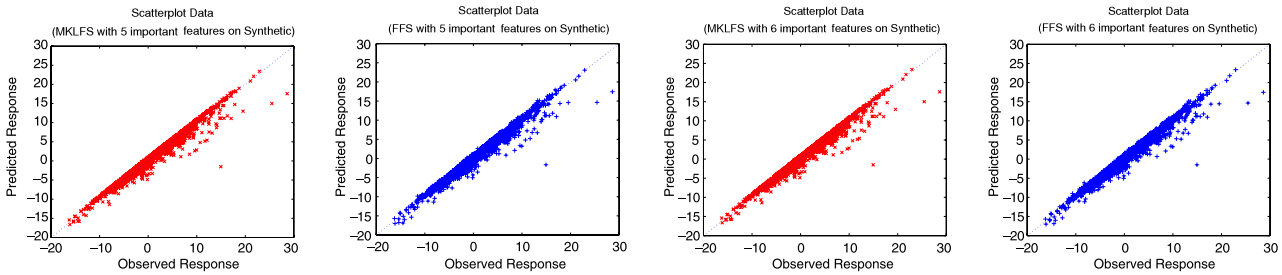
5.1. Experiment on toy data

We first run the experiments on toy dataset. The toy data consisting of $d(= 12)$ dimensions are generated similarly to the additive model in Bi et al. (2003):

$$y_i = \sum_{j=1}^{4} j x_{ji} + e^{x_{5i}}, \tag{22}$$

where y_i denotes the response for the ith sample and $\mathbf{x}_j$ denotes the jth feature, for $j = 1, \ldots, 12$. x_{ji} denotes the element of the jth feature on the ith sample. Here, only the first five features contribute

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



(a) NMMKLR, #SF = 5. (b) Stepwise, #SF = 5. (c) NMMKLR, #SF = 6. (d) Stepwise, #SF = 6.

Fig. 3. Scatter plots of the pairs $(\mathbf{y}, \hat{\mathbf{y}})$. Fig. (a) and Fig. (c) show the plot of NMMKLR when the number of selected features is equal to 5 and 6, respectively. Fig. (b) and Fig. (d) show the plot of Stepwise regression when the number of selected features is equal to 5 and 6, respectively. It can be observed that (a) is thinner than (b) and (c) is thinner than (d).

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.

We conduct two batches of experiments, where the numbers of required features are set up 5 and 6, respectively. Then in each batch of experiment, we randomly generate 200 samples and hold out 50% of the samples for training while keeping the rest for test. Each experiment 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 4. 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 ε, 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 ε is chosen in $[0.01, 0.1, 0.25, 0.5, 0.75, 1, 1.5, 2]$.

We adopt the following two performance measures:

(1) the Mean Square Error (MSE), which is defined as $MSE = \sum_{i=1}^{N}(y_i - \hat{y}_i)^2/N$, where $\hat{y}_i$ is the prediction of y_i for the ith test sample;

(2) the Q^2 statistics, defined as $Q^2 = \frac{\sum_{i=1}^{N}(y_i-\hat{y}_i)^2}{(y_i-\bar{y}_i)^2}$, which is scaled by the variance of the response, where $\bar{y}$ is the mean of the actual response.

Finally, we show the evaluation results on four compared algorithms in Table 5. 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., $\langle \mathbf{y}, \hat{\mathbf{y}} \rangle$, for the NMMKLR and the Stepwise selection method. The results are shown in Fig. 3. 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 Fig. 3 that the proposed NMMKLR has a better performance in both cases.

Table 4
Top 5 and 6 selected features (ordered) from NMMKLR, the forward stepwise feature selection, the greed selection by SVR, the greed selection by LASSO within 20 trials on the synthetic dataset. The stepwise and the LASSO-LW methods include some irrelevant features when the number of selected features is greater than six.

Method	Times	#SF = 5	Times	#SF = 6
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, 6
	10	4, 3, 2, 5, 1	3	4, 3, 5, 2, 6, 1
			2	4, 3, 2, 5, 6, 8 (NV_1)
	6	4, 3, 2, 5, 6	1	3, 4, 2, 5, 6, 1
			1	4, 3, 2, 5, 6, 1
	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)

5.2. Experiment on real-world benchmark datasets

We employ two real-world benchmark datasets, the Boston Housing problem (Harrison & Rubinfeld, 1978), and the Forest Fires dataset (Cortez & Morais, 2007), to evaluate the above four feature selection algorithms.

5.2.1. Experiment on the Boston Housing dataset

The Boston Housing problem (Harrison & Rubinfeld, 1978) is a popular benchmark dataset for evaluating 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 is chosen uniformly from the interval $[10^0, 10^3]$ on a logarithmic scale and ε is chosen from $[0.01, 0.1, 0.5, 1 : 0.5 : 10]$, where the notation of $1 : 0.5 : 10$ is the same definition as the Matlab notation.

Table 5
The test performance (*MSE* and Q^2) on the synthetic dataset evaluated by the four compared 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	**1.1599** $\pm$ 0.6977	**0.0339** $\pm$ 0.0186	1.2156 $\pm$ 0.6893	0.0356 $\pm$ 0.0183
6	**1.1600** $\pm$ 0.6977	**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

Table 6
The results of two performance measures (*MSE* and Q^2) on the Housing dataset when varying the number of selected features by the NMMKLR and the stepwise feature selection, the SVR-LW, and the 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	**25.65** $\pm$ 2.36	**0.3208** $\pm$ 0.0329	26.24 $\pm$ 2.41	0.3281 $\pm$ 0.0326
6	**25.07** $\pm$ 2.50	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

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 6. 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.

5.2.2. Experiment on the forest fires dataset

The Forest Fires dataset (Cortez & Morais, 2007) is a very difficult regression task, whose objective is to predict the burned area of forest fires. It consists of 517 instances with 12 features, such as x/y-axis spatial coordinate within the Montesinho park map, month/date, etc. The response variable is the burned area of the forest. Since some attributes may be correlated, performing features selection is possible to improve the performance of the regression task.

In the experiment, values in the 12 features are normalized in the range of $[-1, 1]$. Since the output response is in the range [0.0, 1090.84] and is very skewed towards 0.0, we transform it by a logarithm, i.e., $\tilde{y}_i = \log(y_i + 1)$. Similarly, we hold out half of the data for training while using the rest data for test. The test is on the log-scaled output ($\tilde{y}$). For SVR, the hyper-parameter, C, is chosen uniformly from the interval $[10^{-5}, 10^{1}]$ on a logarithmic scale and ε is chosen from [0.1, 0.1, 0.5, 1, 1.5, 2].

As the forward stepwise feature selection method only can select few features (less than 4) when the significance level is set to 0.05, for a fair comparison, we test the NMMKLR with $m = 1, 3$. We then calculate the two performance measures, i.e., *MSE* and Q^2, for SVRs trained on the selected features. The results are shown in Table 7. The better results are highlighted according to the paired t-test with 95% confidence level. It can be observed that the regression results by NMMKLR are significantly better than those selected by the forward stepwise feature selection method in both cases, and also outperform the results on SVR-LW, and LASSO-LW.

6. Conclusion

This paper presents a new framework of non-monotonic feature selection that considers the number of selected features during searching for the optimal feature subset. We develop an efficient algorithm via multiple kernel learning 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. Our empirical evaluation on both synthetic datasets and a number of benchmark real-world datasets for the classification and regression tasks shows the promising performance of the proposed framework.

For future work, we aim to employ more efficient optimization techniques to solve large scale non-monotonic feature selection problems. Moreover, it is desirable to extend the current non-monotonic feature selection method to nonlinear feature selection. We leave this as an open problem and our long term goal.

Acknowledgments

The work described in this paper was fully supported by the National Grand Fundamental Research 973 Program of China (No. 2014CB340401), the projects of NSF China (Nos. 61572111, 61440036, 61433014), a 985 project of UESTC (No. A1098531023601041), a Basic Research Project of China Central Universities (No. ZYGX2014J058), the Research Grants Council of the Hong Kong (No. CUHK 413213 and No. CUHK 415113), and Microsoft Research Asia Regional Seed Fund in Big Data Research (Grant No. FY13-RES-SPONSOR-036).

Appendix A. Proof of Theorem 1

Proof. We introduce dual variables $\mathbf{Z}, \alpha$, and s for the LMI constraint in Eq. (7), η_ν and η_δ for $\nu \geq 0$, $\mathbf{q}$ for $\mathbf{p} \geq 0$, and λ for $\mathbf{p}^\top \mathbf{1} = m$. The Lagrangian function is then calculated as:

$$\begin{aligned}\mathcal{L} = {} & t + 2C\delta^\top - \sum_{i=1}^{d} p_i \mathrm{tr}((\mathbf{K}_i \circ \mathbf{y}\mathbf{y}^\top)\mathbf{Z}) - \tau \mathrm{tr}(\mathbf{Z}) \\ & - ts + 2\alpha^\top(\mathbf{1} + \nu - \delta + \theta\mathbf{y}) - \eta_\nu^\top \nu - \eta_\delta^\top \delta \\ & - \mathbf{p}^\top \mathbf{q} - \gamma^\top(\mathbf{1} - \mathbf{p}) + \lambda(\mathbf{p}^\top \mathbf{1} - m).\end{aligned}$$

By setting the first order derivative to be zero, we have

$$s = 1, \qquad \alpha^\top \mathbf{y} = 0, \qquad 2\alpha = \eta_\nu \geq 0, \tag{23}$$

$$\alpha + \eta_\delta = C\mathbf{1}, \tag{24}$$

$$\lambda + \gamma_i - \mathbf{Z}(\mathbf{K}_i \circ \mathbf{y}\mathbf{y}^\top) = q_i \geq 0. \tag{25}$$

We then have the following dual problem:

$$\begin{aligned}\max \quad & -\tau \mathrm{tr}(\mathbf{Z}) + 2\alpha^\top \mathbf{1} - \gamma^\top \mathbf{1} - m\lambda \\ \text{s.t.} \quad & Z \succeq \alpha\alpha^\top, \quad 0 \leq \alpha \leq C, \qquad \alpha^\top \mathbf{y} = 0, \\ & \lambda + \gamma_i \geq \mathrm{tr}(\mathbf{Z}(\mathbf{K}_i \circ \mathbf{y}\mathbf{y}^\top)), \quad i = 1, \ldots, d.\end{aligned} \tag{26}$$

Table 7
The results of two performance measures (*MSE* and Q^2) on the Forest Fires dataset when varying the number of selected features by the NMMKLR and the stepwise feature selection, SVR-LW, and LASSO-LW. 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
1	**1.9662** ± 0.1412	**1.0077** ± 0.0056	2.0046 ± 0.1476	1.0276 ± 0.0295
3	**1.9663** ± 0.1412	**1.0077** ± 0.0057	2.0046 ± 0.1476	1.0276 ± 0.0295
#SF	SVR-LW		LASSO-LW	
	MSE	Q^2	*MSE*	Q^2
1	1.9805 ± 0.1450	1.0149 ± 0.0081	1.9802 ± 0.1416	1.0147 ± 0.0320
3	1.9792 ± 0.1458	1.0142 ± 0.0086	1.9754 ± 0.1655	1.0142 ± 0.0146

Since $\text{tr}(\mathbf{Z}) \geq \text{tr}(\alpha\alpha^\top)$ and $\text{tr}(\mathbf{Z}(\mathbf{K}_i \circ \mathbf{y}\mathbf{y}^\top)) \geq \text{tr}(\alpha\alpha^\top(\mathbf{K}_i \circ \mathbf{y}\mathbf{y}^\top))$, it is clear that setting $\mathbf{Z} = \alpha\alpha^\top$ leads to the maximization of the objective function. Using $\mathbf{Z} = \alpha\alpha^\top$, it is straightforward to simplify Eq. (26) into Eq. (8).

Using the equations from Eqs. (23) to (25) and complementary slackness conditions, we have

$$
\begin{aligned}
&2\alpha = \eta_\nu, \qquad \eta_\nu \circ \nu = 0 \Rightarrow \alpha \circ \nu = 0,\\
&\alpha + \eta_\delta = C\mathbf{1}, \qquad \eta_\delta \circ \delta = 0 \Rightarrow \alpha \circ \delta = C\delta,\\
&\gamma_i(1 - p_i) = 0, \quad i = 1, \ldots, d,\\
&q_i p_i = 0,\ q_i = \lambda + \gamma_i - (\alpha \circ \mathbf{y})^\top \mathbf{K}_i(\alpha \circ \mathbf{y}) \Rightarrow\\
&p_i(\lambda + \gamma_i - (\alpha \circ \mathbf{y})^\top \mathbf{K}_i(\alpha \circ \mathbf{y})) = 0, \quad i = 1, \ldots, d.
\end{aligned}
$$

Using the KKT condition

$$
\begin{pmatrix} \mathbf{K}(\mathbf{p}) \circ (\mathbf{y}\mathbf{y}^\top) + \tau\mathbf{I} & \mathbf{1} + \nu - \delta + \theta\mathbf{y} \\ (\mathbf{1} + \nu - \delta + \theta\mathbf{y})^\top & t \end{pmatrix} \begin{pmatrix} \mathbf{Z} & -\alpha \\ -\alpha^\top & s \end{pmatrix} = \mathbf{0},
$$

$\mathbf{Z} = \alpha\alpha^\top$, and $s = 1$, we have

$$
\begin{pmatrix} \mathbf{K}(\mathbf{p}) \circ (\mathbf{y}\mathbf{y}^\top) + \tau\mathbf{I} & \mathbf{1} + \nu - \delta + \theta\mathbf{y} \\ (\mathbf{1} + \nu - \delta + \theta\mathbf{y})^\top & t \end{pmatrix} \begin{pmatrix} -\alpha \\ 1 \end{pmatrix} = \mathbf{0},
$$

which results in the KKT conditions stated in the theorem. ■

Appendix B. Proof of Corollary 1

Proof. First, we show Eq. (8) can also be rewritten as

$$
\max\ 2\alpha^\top\mathbf{1} - \tau\alpha^\top\alpha - \max_{\mathbf{p}\in\mathcal{D}} \alpha^\top(\mathbf{K}(\mathbf{p}) \circ \mathbf{y}\mathbf{y}^\top)\alpha, \tag{27}
$$

where $\mathcal{D} = \{\mathbf{p} | \mathbf{1}^\top\mathbf{p} = m,\ 0 \leq \mathbf{p} \leq 1\}$. This is because for any $\mathbf{p} \in \mathcal{D}$, we have

$$
m\lambda + \sum_{i=1}^{d} \gamma_i \geq \sum_{i=1}^{d} p_i(\lambda + \gamma_i) \geq \alpha^\top(\mathbf{K}(\mathbf{p}) \circ \mathbf{y}\mathbf{y}^\top)\alpha.
$$

Since $\max_{\mathbf{p}\in\mathcal{D}} \alpha^\top(\mathbf{K}(\mathbf{p}) \circ \mathbf{y}\mathbf{y}^\top)\alpha = \sum_{k=1}^{m} \tau_{i_k}$, we have $m\lambda + \sum_{i=1}^{d} \gamma_i = \sum_{k=1}^{m} \tau_{i_k}$. Since $\tau_{i_k} \leq \lambda + \gamma_{i_k}$, we have $\tau_{i_k} = \lambda + \gamma_{i_k}$, $k = 1, \ldots, m$, which leads to $\lambda \leq \tau_{i_m}$ and $\gamma_{i_k} = 0$ if $k \geq k_{\max}$. Moreover, due to $\lambda + \gamma_{i_k} \geq \tau_{i_k}$ for any $k \geq k_{\max}$, we have $\lambda \geq \tau_{i_{1+k_{\max}}}$. Since $\gamma_{i_k} + \lambda = \tau_{i_k}$ for $k \leq k_{\min}$ and $\lambda \leq \tau_{i_m}$, we have $\gamma_{i_k} > 0$ when $k < k_{\min}$.

Using the KKT conditions in Theorem 1, i.e.,

$$
\begin{aligned}
&0 = \gamma \circ (\mathbf{1} - \mathbf{p}) = 0,\\
&0 = p_i(\lambda + \gamma_i - (\alpha \circ \mathbf{y})^\top \mathbf{K}_i(\alpha \circ \mathbf{y})), \quad i = 1, \ldots, d,
\end{aligned}
$$

we have $p_{i_k} = 1$ if $k < k_{\min}$ and 0 if $k > k_{\max}$, which is the result in Eq. (13). ■

Appendix C. Proof of Theorem 4

Proof. Let $\mathcal{S} \subset \mathcal{B}$ denote the subset of $m + 1 - k_{\min}$ features that are selected by Algorithm 1. We denote by $\mathbf{p}_\mathcal{S}$ the corresponding discrete solution for $\mathbf{p}$. We denote by ν^*, δ^*, and θ^*, the optimal solution to Eq. (7) in addition to $\mathbf{p}^*$.

We first express $\omega(\mathbf{p}_\mathcal{S})$ as an optimization problem, i.e., $\omega(\mathbf{p}_\mathcal{S}) = \min_{\nu\geq 0, \delta\geq 0, \theta} \phi(\nu, \delta, \theta, \mathbf{p}_\mathcal{S})$, where

$$
\begin{aligned}
\phi(\nu, \delta, \theta, \mathbf{p}) = {} & 2C\delta^\top\mathbf{1} + (\mathbf{1} + \nu - \delta + \theta\mathbf{y})^\top(\mathbf{K}(\mathbf{p}) \circ \mathbf{y}\mathbf{y}^\top + \tau\mathbf{I})^{-1}\\
& \times (\mathbf{1} + \nu - \delta + \theta\mathbf{y}).
\end{aligned}
$$

We upper bound $\omega(\mathbf{p}_\mathcal{S})$ by substituting $(\nu, \delta, \theta) = (\nu^*, \delta^*, \theta^*)$, i.e., $\omega(\mathbf{p}_\mathcal{S}) \leq \phi(\nu^*, \delta^*, \theta^*, \mathbf{p}_\mathcal{S})$. Using the KKT conditions in Theorem 1, we have

$$
\mathbf{1} + \nu^* - \delta^* + \theta^*\mathbf{y} = (\mathbf{K}(\mathbf{p}^*) \circ \mathbf{y}\mathbf{y}^\top + \tau\mathbf{I})\alpha = \mathbf{M}\alpha.
$$

We then focus on bounding

$$
t = (\mathbf{M}\alpha)^\top(\mathbf{K}(\mathbf{p}_\mathcal{S}) \circ \mathbf{y}\mathbf{y}^\top + \tau\mathbf{I})^{-1}\mathbf{M}\alpha.
$$

Based on the Schur complement, we have

$$
\begin{aligned}
&t = \arg\min_z z\\
&\text{s.t.} \begin{pmatrix} \mathbf{K}(\mathbf{p}_\mathcal{S}) \circ \mathbf{y}\mathbf{y}^\top + \tau\mathbf{I} & \mathbf{M}\alpha \\ (\mathbf{M}\alpha)^\top & z \end{pmatrix} \succeq 0.
\end{aligned} \tag{28}
$$

We rewrite $\mathbf{K}(\mathbf{p}_\mathcal{S}) \circ \mathbf{y}\mathbf{y}^\top + \tau\mathbf{I}$ as

$$
\mathbf{K}(\mathbf{p}_\mathcal{S}) \circ \mathbf{y}\mathbf{y}^\top + \tau\mathbf{I} = \mathbf{M} + \mathbf{A}_\mathcal{S} - \mathbf{B},
$$

where

$$
\mathbf{A}_\mathcal{S} = \sum_{j\in\mathcal{S}} \mathbf{K}_j \circ \mathbf{y}\mathbf{y}^\top, \qquad \mathbf{B} = \sum_{j\in\mathcal{B}} p_j^* \mathbf{K}_j \circ \mathbf{y}\mathbf{y}^\top.
$$

We introduce a parameter ρ, and relax the condition in Eq. (28) as follows:

$$
\begin{aligned}
&\begin{pmatrix} (1-\rho)(\mathbf{M} + \mathbf{A}_\mathcal{S} - \mathbf{B}) & (\mathbf{M} + \mathbf{A}_\mathcal{S} - \mathbf{B})\alpha \\ ((\mathbf{M} + \mathbf{A}_\mathcal{S} - \mathbf{B})\alpha)^\top & t_1 \end{pmatrix} \succeq 0,\\
&\begin{pmatrix} \rho(\mathbf{M} + \mathbf{A}_\mathcal{S} - \mathbf{B}) & (\mathbf{B} - \mathbf{A}_\mathcal{S})\alpha \\ ((\mathbf{B} - \mathbf{A}_\mathcal{S})\alpha)^\top & t_2 \end{pmatrix} \succeq 0,\\
&t_1 + t_2 \geq t.
\end{aligned}
$$

We thus have

$$
t \leq \frac{\alpha^\top\mathbf{M}\alpha}{1-\rho} - \alpha^\top\left[(\mathbf{B} - \mathbf{A}_\mathcal{S})(\mathbf{M} + \mathbf{A}_\mathcal{S} - \mathbf{B})^{-1}\mathbf{V}\right]\alpha,
$$

where

$$
\mathbf{V} = \frac{1}{1-\rho}\left(\mathbf{M} - \frac{\mathbf{B} - \mathbf{A}_\mathcal{S}}{\rho}\right).
$$

By choosing $\rho^* = \sigma_{\max}(\mathbf{M}^{-1/2}\mathbf{B}\mathbf{M}^{-1/2}) \leq 1$ where the operator $\sigma_{\max}(\cdot)$ calculates the maximal eigenvalue, we have $\mathbf{V} \succeq 0$ and $t(1-\rho^*) \leq \alpha^\top \mathbf{M}\alpha$. Thus, the performance bound is

$$\frac{\omega(\mathbf{p}_\delta)}{\omega(\widetilde{\mathbf{p}}^*)} \leq \frac{\omega(\mathbf{p}_\delta)}{\omega(\mathbf{p}^*)} \leq \frac{t + 2C\mathbf{1}^\top\delta^*}{\alpha^\top\mathbf{M}\alpha + 2C\mathbf{1}^\top\delta^*} \leq \frac{1}{1-\rho^*}. \quad \blacksquare$$

References

Augasta, M. G., & Kathirvalavakumar, T. (2013). Pruning algorithms of neural networks—a comparative study. *Central European Journal of Computer Science*, *3*(3), 105–115.

Bach, F.R., Lanckriet, G.R.G., & Jordan, M.I. (2004). Multiple kernel learning, conic duality, and the SMO algorithm. In *ICML'04*.

Bhattacharyya, C. (2004). Second order cone programming formulations for feature selection. *Journal of Machine Learning Research*, *5*, 1417–1433.

Bi, J., Bennett, K. P., Embrechts, M. J., Breneman, C. M., & Song, M. (2003). Dimensionality reduction via sparse support vector machines. *Journal of Machine Learning Research*, *3*, 1229–1243.

Bishop, C. M. (1996). *Neural networks for pattern recognition*. Oxford University Press.

Blum, A., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, *97*(1–2), 245–271.

Boutsidis, C., Mahoney, M.W., & Drineas, P. (2009). Unsupervised feature selection for the *k*-means clustering problem. In *NIPS'09* (pp. 153–161).

Bradley, P.S., & Mangasarian, O.L. (1998). Feature selection via concave minimization and support vector machines. In *ICML'98* (pp. 82–90).

Chan, A.B., Vasconcelos, N., & Lanckriet, G.R.G. (2007). Direct convex relaxations of sparse SVM. In *ICML'07* (pp. 145–153).

Cortez, P., & Morais, A. (2007). A data mining approach to predict forest fires using meteorological data. In *J. Neves, M. F. Santos, and J. Machado, (Eds.), Proceedings of the 13th EPIA 2007—Portuguese conference on artificial intelligence* (pp. 512–523).

Cristianini, N., Shawe-Taylor, J., Elisseeff, A., & Kandola, J.S. (2001). On kernel-target alignment. In *NIPS'01* (pp. 367–373).

Dekel, O., & Singer, Y. (2006). Support vector machines on a budget. In *NIPS'06* (pp. 345–352).

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. *Annals of Statistics*, *32*(2), 407–499.

Fisher, D. H. (1996). Iterative optimization and simplification of hierarchical clusterings. *Journal of Artificial Intelligence Research (JAIR)*, *4*, 147–178.

Fung, G., & Mangasarian, O.L. (2000). Data selection for support vector machine classifiers. In *SIGKDD* (pp. 64–70).

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*, 1157–1182.

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

Harrison, D. J., & Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, *5*(1), 81–102.

He, X., & Niyogi, P. (2003). Locality preserving projections. In *NIPS'03* (pp. 153–160).

Hsuen Chen, P., Lin, C.-J., & Schölkopf, B. (2005). A tutorial on ν-support vector machines. *Applied Stochastic Models in Business and Industry*, *21*(2), 111–136.

Huang, K., King, I., & Lyu, M.R. (2008). Direct zero-norm optimization for feature selection. In *ICDM* (pp. 845–850).

Huang, K., Yang, H., King, I., & Lyu, M. R. (2008). *Advanced topics in science and technology in China: machine learning, Modeling data locally and globally*. Zhejiang University Press with Springer Verlag.

Ivakhnenko, A.G. (1995). The review of problems solvable by algorithms of the group method of data handling (GMDH).

Jolliffe, I. (1986). *Principal component analysis*. New York: Springer-Verlag.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*(1–2), 273–324.

Kohonen, T. (2006). Self-organizing neural projections. *Neural Networks*, *19*(6–7), 723–733.

Koller, D., & Sahami, M. (1996). Toward optimal feature selection. In *ICML'96* (pp. 284–292).

Lanckriet, G. R. G., Cristianini, N., Bartlett, P. L., Ghaoui, L. E., & Jordan, M. I. (2004). Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, *5*, 27–72.

Malhi, A., & Gao, R. X. (2004). Pca-based feature selection scheme for machine defect classification. *IEEE Transactions on Instrumentation and Measurement*, *53*(6), 1517–1525.

Margineantu, D., Greiner, R., Singliar, T., & Melville, P. (2010). In *ICML workshop on budgeted learning*.

Neumann, J., Schnörr, C., & Steidl, G. (2005). Combined SVM-based feature selection and classification. *Machine Learning*, *61*(1–3), 129–150.

Ng, A.Y. (2004). Feature selection, L1 vs. L2 regularization, and rotational invariance. In *ICML'04*, (pp. 78–86).

Rakotomamonjy, A. (2003). Variable selection using svm-based criteria. *Journal of Machine Learning Research*, *3*, 1357–1370.

Reddy, K.N., & Ravi, V. (2012). Kernel group method of data handling: Application to regression problems. In *SEMCCO* (pp. 74–81).

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*(3), 199–222.

Song, L., Smola, A.J., Gretton, A., Borgwardt, K.M., & Bedo, J. (2007). Supervised feature selection via dependence estimation. In *ICML'07* (pp. 823–830).

Sonnenburg, S., Rätsch, G., Schäfer, C., & Schölkopf, B. (2006). Large scale multiple kernel learning. *Journal of Machine Learning Research*, *7*, 1531–1565.

Tan, M., Tsang, I. W., & Wang, L. (2014). Towards ultrahigh dimensional feature selection for big data. *Journal of Machine Learning Research*, *15*(1), 1371–1429.

Thi, H. A. L., Vo, X. T., & Dinh, T. P. (2014). Feature selection for linear svms under uncertain data: Robust optimization based on difference of convex functions algorithms. *Neural Networks*, *59*, 36–50.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B*, *58*(1), 267–288.

Vapnik, V. (1998). *Statistical learning theory*. John Wiley & Sons.

Vapnik, V. (1999). *The nature of statistical learning theory* (2nd ed.). New York: Springer.

Wang, J. J., Bensmail, H., & Gao, X. (2014). Feature selection and multi-kernel learning for sparse representation on a manifold. *Neural Networks*, *51*, 9–16.

Wang, J., Zhao, P., Hoi, S. C. H., & Jin, R. (2014). Online feature selection and its applications. *IEEE Transactions on Knowledge and Data Engineering*, *26*(3), 698–710.

Weston, J., Elisseeff, A., Schölkopf, B., & Tipping, M. E. (2003). Use of the zero-norm with linear models and kernel methods. *Journal of Machine Learning Research*, *3*, 1439–1461.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., & Vapnik, V. (2000). Feature selection for SVMs. In *NIPS'00* (pp. 668–674).

Wolf, L., & Shashua, A. (2005). Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach. *Journal of Machine Learning Research*, *6*, 1855–1887.

Xu, Z., Jin, R., Ye, J., Lyu, M.R., & King, I. (2009). Non-monotonic feature selection. In *ICML'09* (pp. 1145–1152).

Xu, Z., King, I., & Lyu, M. R. (2007). Feature selection based on minimum error minimax probability machine. *International Journal of Pattern Recognition and Artificial Intelligence*, *21*(8), 1279–1292.

Xu, Z., King, I., Lyu, M. R., & Jin, R. (2010). Discriminative semi-supervised feature selection via manifold regularization. *IEEE Transactions on Neural Networks*, *21*(7), 1033–1047.

Yang, H., Chan, L., & King, I. (2002). Support vector machine regression for volatile stock market prediction. In *IDEAL* (pp. 391–396).

Yang, H., Huang, K., King, I., & Lyu, M. R. (2009). Localized support vector regression for time series prediction. *Neurocomputing*, *72*(10–12), 2659–2669.

Yang, H., King, I., & Lyu, M. R. (2011). *Sparse learning under regularization framework*. LAP Lambert Academic Publishing.

Yang, H., Lyu, M. R., & King, I. (2013). Efficient online learning for multi-task feature selection. *ACM Transactions on Knowledge Discovery from Data*, *7*(2), 1–27.

Yang, H., Xu, Z., King, I., & Lyu, M.R. (2010). Online learning for group lasso. In *ICML, Haifa, Israel* (pp. 1191–1198).

Yang, H., Xu, Z., King, I., & Lyu, M.R. (2014). Non-monotonic feature selection for regression. In *ICONIP (2)* (pp. 44–51).

Yang, H., Xu, Z., Ye, J., King, I., & Lyu, M. R. (2011). Efficient sparse generalized multiple kernel learning. *IEEE Transactions on Neural Networks*, *22*(3), 433–446.

Yu, L., Ding, C.H.Q., & Loscalzo, S. (2008). Stable feature selection via dense feature groups. In *SIGKDD* (pp. 803–811).

Zhao, Z., & Liu, H. (2007). Spectral feature selection for supervised and unsupervised learning. In *ICML'07* (pp. 1151–1157).