Non-Monotonic Feature Selection

Zenglin Xu ZLXU@CSE.CUHK.EDU.HK
Department of Computer Science & Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong

Rong Jin RONGJIN@CSE.MSU.EDU
Department of Computer Science & Engineering, Michigan State University, East Lansing, MI 48824 USA

Jieping Ye JIEPING.YE@ASU.EDU
Department of Computer Science & Engineering, Arizona State University, Tempe, AZ 85287-8809, USA

Michael R. Lyu, Irwin King {LYU,KING}@CSE.CUHK.EDU.HK
Department of Computer Science & Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong

Abstract

We consider the problem of selecting a subset of m most informative features where m is the number of required features. This feature selection problem is essentially a combinatorial optimization problem, and is usually solved by an approximation. Conventional feature selection methods address the computational challenge in two steps: (a) ranking all the features by certain scores that are usually computed independently from the number of specified features m, and (b) selecting the top m ranked features. One major shortcoming of these approaches is that if a feature f is chosen when the number of specified features is m, it will always be chosen when the number of specified features is larger than m. We refer to this property as the "*monotonic*" property of feature selection. In this work, we argue that it is important to develop efficient algorithms for non-monotonic feature selection. To this end, we develop an algorithm for non-monotonic feature selection that approximates the related combinatorial optimization problem by a Multiple Kernel Learning (MKL) problem. We also present a strategy that derives a discrete solution from the approximate solution of MKL, and show the performance guarantee for the derived discrete solution when compared to the global optimal solution for the related combinatorial optimization problem. An empirical study with a number of benchmark data sets indicates the promising performance of the proposed framework compared with several state-of-the-art approaches for feature selection.

Appearing in *Proceedings of the 26^{th} International Conference on Machine Learning*, Montreal, Canada, 2009.

1. Introduction

Feature selection is an important task in machine learning and has been studied extensively. 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. It has also been used for data visualization. Feature selection has found applications in a number of real-world problems, such as natural language processing, computer vision, bioinformatics, and sensor networks.

One of the important issues in feature selection is to set the number of required features. It is important to note that determining the number of selected features is a model selection problem, and is beyond the scope of this study. In this work, we assume that an external oracle decides the number of selected features. It should also be noted that the number of required features usually depends on the objective of the task, and there is no single number of features that are optimal for all tasks.

Given the number of required features, 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:

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

A number of performance criteria have been proposed for feature selection, including mutual information (Koller &

Sahami, 1996), maximum margin (Weston et al., 2000; Guyon et al., 2002), kernel alignment (Cristianini et al., 2001; Neumann et al., 2005), and the Hilbert Schmidt independence criterion (Song et al., 2007), etc. Among them, the maximum-margin-based criterion is probably one of the most widely used criteria for feature selection due to its outstanding performance.

The computational challenge in solving the optimization problem in (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 (1). Most of them first compute a weight/score w for each feature, and then select features with the largest weights. 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 & Mangasarian, 2000; Ng, 2004), L_1-norm of weights was suggested to replace L_2-norm for feature selection when learning an SVM model. Another feature selection model related to L_1-norm is lasso (Tibshirani, 1996), which selects features by constraining the L_1-norm of weights. By varying L_1-norm of weights, a unique path of selected features can be obtained. A similar model is LARS (Efron et al., 2004), which can be regarded as unconstrained version of lasso. In addition to the optimization on L_2-norm and L_1-norm, several studies (Bradley & Mangasarian, 1998; Weston et al., 2003; Neumann et al., 2005; Chan et al., 2007) explored L_0-norm when computing the weights of features. In (Bradley & Mangasarian, 1998), the authors proposed Feature Selection Concave method (FSV) that uses an approximate L_0-norm of the weights. It was improved in (Weston et al., 2003; Neumann et al., 2005) via an additional regularizer or a different approximation of L_0-norm. In addition to selecting features by weights, in (Vapnik, 1998; Weston et al., 2000; Rakotomamonjy, 2003), 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.

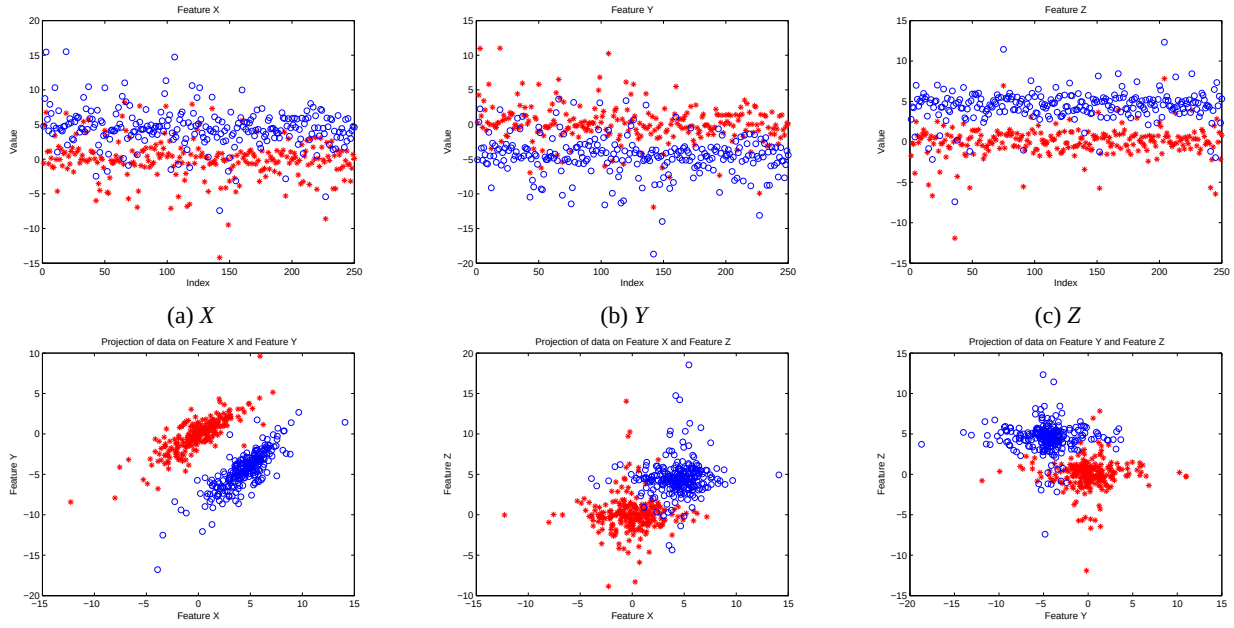
Although the above approximate approaches have been successfully applied to a number of applications of feature selection, they are limited by the **monotonic** property of feature selection that is defined below:

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 $\mathcal{A}$.

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 & 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 bi-category 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 (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 (1) into a formulation that is closely related to multiple kernel learning (MKL) (Lanckriet et al., 2004; Bach et al., 2004; Sonnenburg et al., 2006; Rakotomamonjy et al., 2007; Xu et al., 2009; Cortes et al., 2008). 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. 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 state-of-the-arts 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



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

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

Figure 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 $\circ$ and $*$, respectively.

the non-monotonic feature selection in Section 2. Section 3 presents experimental results with a number of benchmark data sets. We conclude our work in Section 4.

2. Non-monotonic Feature Selection via Multiple Kernel Learning

In this section, we first show that multiple kernel learning approaches 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 the feature index set $\{i\}_{i=1}^d$ as $\mathcal{P}$. We denote $\mathbf{e}_d \in \mathbf{R}^d$ as a d-dimensional vector with all elements being one. We also omit the suffix when the dimensionality d of $\mathbf{e}_d$ can be easily inferred from the context. 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. 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 (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:

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

where $\mathbf{I}$ is the identity matrix; α is the dual variable for the margin error; both C and τ are manually set constants; $\circ$ 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 $\mathbf{p} = \mathbf{e}$, then (3) reduces to a standard SVM.

We denote by $\omega(\mathbf{p})$ the value of the objective function in (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.,

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

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

optimization problem:

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

Remark It is important to note that although the objective function in (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, (5) may have a non-discrete solution. To see this, consider the problem

$$\min_{0\leq\mathbf{p}\leq 1,\mathbf{p}^\top\mathbf{e}=1} \quad \max_{\mathbf{x}\in\mathbb{R}^d} \quad 2\mathbf{p}^\top\mathbf{x} - \|\mathbf{x}\|_2^2. \tag{6}$$

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

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

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

$$\begin{aligned}\min_{\mathbf{p},t,\nu,\delta,\theta} \quad & t + 2C\delta^\top\mathbf{e} \\ \text{s. t.} \quad & \begin{pmatrix} \mathbf{K}(\mathbf{p})\circ(\mathbf{y}\mathbf{y}^\top) + \tau\mathbf{I} & \mathbf{e}+\nu-\delta+\theta\mathbf{y} \\ (\mathbf{e}+\nu-\delta+\theta\mathbf{y})^\top & t \end{pmatrix} \succeq 0, \\ & \nu\geq 0,\ \delta\geq 0,\ \mathbf{p}^\top\mathbf{e} = m,\ 0\leq\mathbf{p}\leq 1.\end{aligned} \tag{7}$$

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

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

$$\begin{aligned}\max_{\alpha,\lambda,\gamma} \quad & 2\alpha^\top\mathbf{e} - \tau\alpha^\top\alpha - m\lambda - \gamma^\top\mathbf{e} \\ \textit{s. t.} \quad & \alpha^\top\mathbf{y} = 0,\ 0\leq\alpha\leq C, \\ & (\alpha\circ\mathbf{y})^\top\mathbf{K}_i(\alpha\circ\mathbf{y}) \leq \lambda+\gamma_i,\ \forall i\in\mathcal{P}, \\ & \gamma_i\geq 0,\ \forall i\in\mathcal{P}.\end{aligned} \tag{8}$$

The KKT conditions are

$$\begin{aligned}&(\mathbf{K}(\mathbf{p})\circ\mathbf{y}\mathbf{y}^\top + \tau\mathbf{I})\alpha = \mathbf{e}+\nu-\delta+\theta\mathbf{y}, \\ &t = \alpha^\top(\mathbf{e}+\nu-\delta+\theta\mathbf{y}), \\ &\nu\circ\alpha = 0,\ \alpha\circ\delta = C\delta,\ \gamma\circ(\mathbf{e}-\mathbf{p}) = 0, \\ &p_i(\lambda+\gamma_i - (\alpha\circ\mathbf{y})^\top\mathbf{K}_i(\alpha\circ\mathbf{y})) = 0, \forall i\in\mathcal{P}.\end{aligned}$$

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

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

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

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

$$\begin{aligned}\max_{\alpha,\lambda,\gamma} \quad & 2\alpha^\top\mathbf{e} - \tau\alpha^\top\alpha + \lambda(d-m) - \sum_{i=1}^{d}\max(\lambda,\tau_i) \\ \text{s. t.} \quad & \alpha^\top\mathbf{y} = 0,\ 0\leq\alpha\leq C,\ \lambda\geq 0,\ \gamma\geq 0.\end{aligned} \tag{14}$$

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 (14), we have the formulation of a linear SVM in (3). □

Remark The desired number of selected features, i.e., m, controls the sparseness of features. It is related to the ν-SVM (Schölkopf et al., 2000), which bounds the ratio of support vectors.

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

Theorem 4. *Given the solution to the dual problem in (8), denoted by α, γ, and λ, the solution to the primal problem in (7) can be found by solving the following linear programming problem:*

$$\begin{aligned}\min_{\mathbf{p},\nu,\delta} \quad & \alpha^\top(\mathbf{K}(\mathbf{p})\circ\mathbf{y}\mathbf{y}^\top + \tau\mathbf{I})\alpha + 2C\mathbf{e}^\top\delta \\ \textit{s. t.} \quad & (\mathbf{K}(\mathbf{p})\circ\mathbf{y}\mathbf{y}^\top + \tau\mathbf{I})\alpha = \mathbf{e}+\nu-\delta+\theta\mathbf{y}, \\ & \nu\circ\alpha = 0,\ \alpha\circ\delta = C\delta,\ \delta\geq 0,\ \nu\geq 0, \\ & 0\leq\mathbf{p}\leq 1,\ \mathbf{e}^\top\mathbf{p} = m,\ \gamma\circ(\mathbf{e}-\mathbf{p}) = 0, \\ & p_i(\lambda+\gamma_i - (\alpha\circ\mathbf{y})^\top\mathbf{K}_i(\alpha\circ\mathbf{y})) = 0,\ \forall i\in\mathcal{P}.\end{aligned} \tag{15}$$

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

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

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

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

Theorem 5. *Consider the following problem:*

$$\begin{aligned} \min_{\mathbf{p},\nu,\delta} \quad & \alpha^\top(\mathbf{K}(\mathbf{p}) \circ \mathbf{y}\mathbf{y}^\top + \tau\mathbf{I})\alpha + 2C\mathbf{e}^\top\delta \quad (17) \\ \text{s. t.} \quad & \nu \circ \alpha = 0,\ \alpha \circ \delta = C\delta,\ \delta \geq 0,\ \nu \geq 0, \\ & 0 \leq \mathbf{p} \leq 1,\ \mathbf{e}^\top\mathbf{p} = m,\ \gamma \circ (\mathbf{e} - \mathbf{p}) = 0, \\ & p_i(\lambda + \gamma_i - (\alpha \circ \mathbf{y})^\top\mathbf{K}_i(\alpha \circ \mathbf{y})) = 0,\ \forall i \in \mathcal{P}, \end{aligned}$$

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

Proof. First, notice that an optimal solution for δ and ν to (17) is $\delta = \nu = 0$. Since (13) gives binary solutions for p_i if $i \in \mathcal{A} \cup \mathcal{C}$, the only remaining undecided variables for (17) are $\{p_i | i \in \mathcal{B}\}$. Second, notice that the objective function in (17) remains the same no matter which subset of $s = m + 1 - k_{\min}$ features are selected from $\mathcal{B}$. This 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 (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.

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 (8), not by (3).

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

Theorem 6. *The discrete solution constructed by Algorithm 1, denoted by $\mathbf{p}$, has the following performance guarantee for the combinatorial optimization problem in (4):*

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

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.

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 (5), and $\widetilde{\mathbf{p}}^*$ is the global optimal solution to the combinatorial optimization problem in (4).

The proof can be found in the long version of this paper. As indicated by Theorem 6, 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 (4).

3. Experiment

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 state-of-the-art approaches for feature selection:

- **Fisher** (Bishop, 1995) that calculates a Fisher/Correlation score for each feature.
- **FSV** (Bradley & Mangasarian, 1998) that approximates the L_0-norm of $\mathbf{w}$ by a summation of exponential functions.
- R^2W^2 (Weston et al., 2000) that adjusts weight $\mathbf{w}$ by computing gradient descents on a bound of the leave-one-out error.
- L_0-**appr** (Weston et al., 2003) that approximates the L_0-norm by minimizing a logarithm function.
- L_1-**SVM** (Fung & Mangasarian, 2000) that replaces L_2-norm of $\mathbf{w}$ with 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 & Mangasarian, 2000); for other baseline algorithms, we adopt the implementations in Spider (`www.kyb.tuebingen.mpg.de/bs/people/spider/`).

Table 1. The test accuracy (%) for the toy data set. #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

3.1. Experiment on Toy Data

We first run our experiments over the toy dataset that is illustrated in Fig. 1. We randomly select 400 examples from the toy 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 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.

3.2. Experiment on Real-World Data Sets

The data sets well studied from previous literatures of feature selection (Guyon et al., 2002; Weston et al., 2003) are employed in our experiments. We select data sets from three different data repositories for our evaluation: (a) four binary data sets from the UCI repository (`http://archive.ics.uci.edu/ml/`), namely Ionosphere, Sonar, Wdbc, and Wpbc; (b) three data sets from the Semi-supervised Learning book (`www.kyb.tuebingen.mpg.de/ssl-book/`), namely Digit1, Usps, and Bci; and (c) two microarray data sets (`www.kyb.tuebingen.mpg.de/bs/people/weston/l0/`), namely Colon and Lymphoma. Table 2 lists the size for each data set.

Note that the two microarray data sets are rather challenging compared to the other data sets since they contain a small number of data points but have very high dimensionalities. 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 data sets, 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 toy data set, is also applied to the nine real-world data sets to normalize data and decide parameters C and τ. To speed up the computation for the two microarray data sets (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 data sets, and 10, 20, 40, and 60 for the other five data sets. This is because Bci, Digit1, Usps, and the two micro-array data sets contain examples with significantly higher dimensionality than the UCI data sets, and therefore allow for larger numbers of selected features.

We present the classification results for the four UCI data sets in Table 3 and the results of the remaining data sets in Figure 2.[1] 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 data sets with different number of selected features (e.g., Colon and Sonar with 10 and 20 features), the improvement is significant. As revealed in Corollary 3, 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 data sets. We find that overall the proposed approach performs slightly better than L_0-appr and L_1-SVM for most of the cases. For data sets Sonar and Bci, the improvement made by the proposed algorithm is statistically significant

[1] Since R^2W^2 and FSV are time consuming on high dimensional data sets, we do not include their results.

Table 2. Data sets 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

(student-t) 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 data sets 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 data sets. 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 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 data sets than the others. This may be attributed by the very high dimensions of the two data sets.

4. 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 study with a number of benchmark data sets 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. We also plan to study the tightness of the performance guarantee stated in Theorem 6. 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.

Acknowledgement

The work was supported by the National Science Foundation (IIS-0643494), National Institute of Health (1R01GM079688-01) and Research Grants Council of Hong Kong (CUHK4158/08E and CUHK4128/08E). This work is also affiliated with the Microsoft-CUHK Joint Laboratory for Human-centric Computing and Interface Technologies.

References

Bach, F. R., Lanckriet, G. R. G., & Jordan, M. I. (2004). Multiple kernel learning, conic duality, and the SMO algorithm. *Proc. Int. Conf. Mach. Learn.* (pp. 41–48).

Bishop, C. (1995). *Neural networks for pattern recognition*. London: Oxford University Press.

Bradley, P. S., & Mangasarian, O. L. (1998). Feature selection via concave minimization and support vector machines. *Proc. Int. Conf. Mach. Learn.* (pp. 82–90).

Chan, A. B., Vasconcelos, N., & Lanckriet, G. R. G. (2007). Direct convex relaxations of sparse svm. *Proc. Int. Conf. Mach. Learn.* (pp. 145–153).

Cortes, C., Mohri, M., & Rostamizadeh, A. (2008). Learning sequence kernels. *IEEE Workshop on Machine Learning for Signal Processing* (pp. 2–8).

Cristianini, N., Shawe-Taylor, J., Elisseeff, A., & Kandola, J. S. (2001). On kernel-target alignment. *Proc. Neur. Info. Proc. Sys.* (pp. 367–373).

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

Fung, G., & Mangasarian, O. L. (2000). Data selection for support vector machine classifiers. *Proc. Know. Disc. and Data Min.* (pp. 64–70).

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *J. Mach. Learn. Res., 3*, 1157–1182.

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

Koller, D., & Sahami, M. (1996). Toward optimal feature selection. *Proc. Int. Conf. Mach. Learn.* (pp. 284–292).

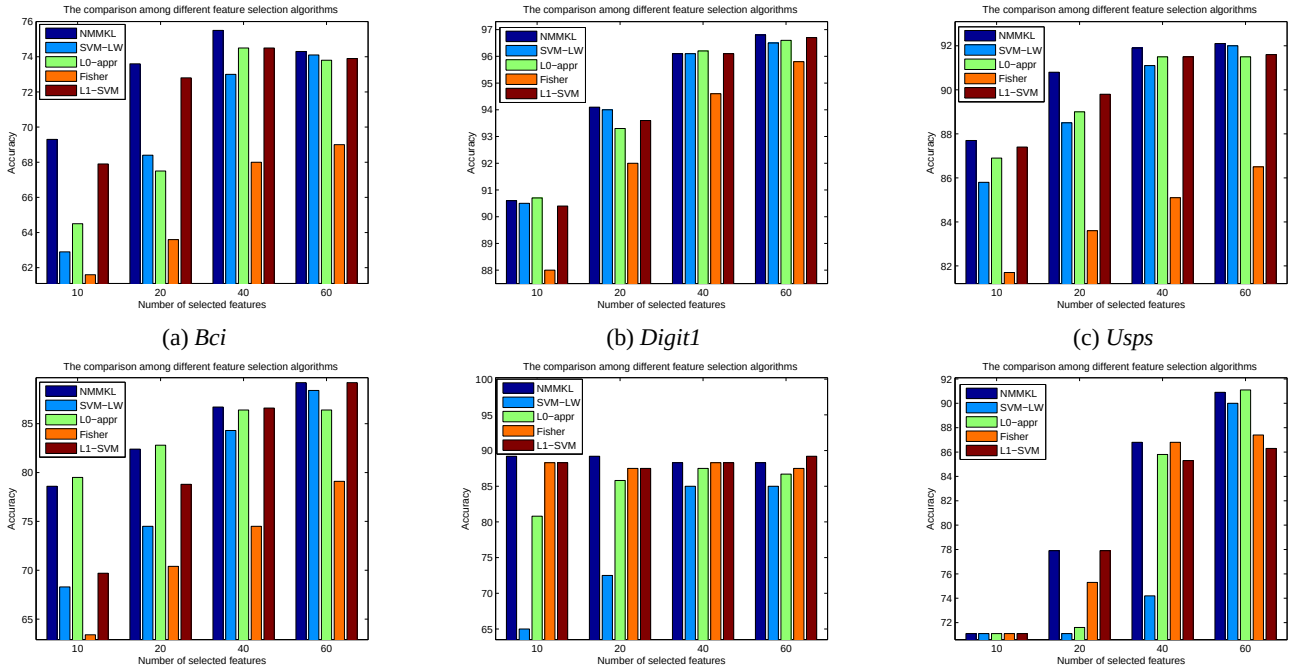
Lanckriet, G. R. G., Cristianini, N., Bartlett, P., Ghaoui, L. E., & Jordan, M. I. (2004). Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res., 5*, 27–72.

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

Ng, A. Y. (2004). Feature selection, L1 vs. L2 regularization, and rotational invariance. *Proc. Int. Conf. Mach. Learn.* (pp. 78–86).

Table 3. The classification accuracy (%) on real-world data sets. 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*

Figure 2. The classification accuracy of feature selection algorithms

Rakotomamonjy, A. (2003). Variable selection using svm-based criteria. *J. Mach. Learn. Res.*, *3*, 1357–1370.

Rakotomamonjy, A., Bach, F., Canu, S., & Grandvalet, Y. (2007). More efficiency in multiple kernel learning. *Proc. Int. Conf. Mach. Learn.* (pp. 775–782).

Schölkopf, B., Smola, A. J., Williamson, R. C., & Bartlett, P. L. (2000). New support vector algorithms. *Neural Computation*, *12*, 1207–1245.

Song, L., Smola, A., Gretton, A., Borgwardt, K. M., & Bedo, J. (2007). Supervised feature selection via dependence estimation. *Proc. Int. Conf. Mach. Learn.* (pp. 823–830).

Sonnenburg, S., Rätsch, G., Schäfer, C., & Schölkopf, B. (2006). Large scale multiple kernel learning. *J. Mach. Learn. Res.*, *7*, 1531–1565.

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

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

Weston, J., Elisseeff, A., Schölkopf, B., & Tipping, M. (2003). Use of the zero norm with linear models and kernel methods. *J. Mach. Learn. Res.*, *3*, 1439–1461.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., & Vapnik, V. (2000). Feature selection for SVMs. *Proc. Neur. Info. Proc. Sys.* (pp. 668–674).

Xu, Z., Jin, R., King, I., & Lyu, M. R. (2009). An extended level method for efficient multiple kernel learning. *Proc. Neur. Info. Proc. Sys.* (pp. 1825–1832).