Appl Intell (2018) 48:1776–1790
DOI 10.1007/s10489-017-1054-5

New fast feature selection methods based on multiple support vector data description

Li Zhang[1,2] · **Xingning Lu**[1]

Published online: 7 September 2017

Abstract Feature selection can sort out useful features to obtain good performance when dealing with high-dimensional data. Feature selection methods based on support vector data description (SVDD) have been proposed for one-class classification problems: SVDD-radius-recursive feature elimination (SVDD-RRFE) and SVDD-dual-objective-recursive feature elimination (SVDD-DRFE). However, both SVDD-RRFE and SVDD-DRFE use only one-class samples even given a multi-class classification task, and suffer from high computational complexity. To remedy it, this paper extends both SVDD-RRFE and SVDD-DRFE to binary and multi-class classification problems using multiple SVDD models, and proposes fast feature ranking schemes for them in the case of the linear kernel. Experimental results on toy, UCI and microarray datasets show the efficiency and the feasibility of the proposed methods.

Keywords Feature selection · Classification problems · Support vector data description · Multiple models

✉ Li Zhang
zhangliml@suda.edu.cn

[1] School of Computer Science and Technology, Joint International Research Laboratory of Machine Learning and Neuromorphic Computing, Soochow University, Suzhou 215006, Jiangsu, China

[2] Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210023, Jiangsu, China

1 Introduction

Classification problems exist extensively in human activities. People can differentiate various groups of objects by their features. Sometimes features are too many to neatly distinguish an object since redundant features would confuse human cognition. Advanced artificial technologies can be introduced into a classification process. Even so, information acquisition demands the extra cost of space and time. In that case, an approach like feature selection can assist in simplifying the features. Feature selection is frequently applied to classification problems with high-dimensional data in which each feature denotes a separate dimension [12, 35]. For high-dimensional data, features involve three categories: relevant, irrelevant and redundant [22]. Feature selection aims at obtaining a relevant feature subset by eliminating irrelevant and redundant features from given original data. Compared with utilizing all features, feature selection can achieve a similar or superior classification performance and have a lower computational complexity [2, 14, 27].

Dash and Liu divided feature selection methods into four fundamental processes: generation procedure, evaluation function, stopping criterion and validation procedure [6]. The first two processes count as the key ones among the four processes. The generation procedure can be one of three types, complete, heuristic and random generation [17, 25]. With the advantages of fast setting up models and simple implementation, the heuristic generation is often broached, such as sequential backward selection (SBS) [23] and sequential forward selection (SFS) [20].

The evaluation function could be a function of the measurement of distance, information, dependency, consistency, or classifier error rate on data, where the last measurement is

frequently used in SBS, SFS and their extensions. Based on the evaluation function, two methods regularly appear in the researches of feature selection: wrapper and filter methods [6, 21]. The approaches using the classification error rate as the evaluation function are referred to wrapper ones. Alternatively, filter methods pick up features via statistics, for instance, the mean or variance of measurements. Generally speaking, wrapper methods can gain advanced classification performance compared to filter methods. The reason is that wrapper methods work with the assist of selectable classifiers and filter methods ignore classifiers when processing data. However, at the cost of the classification accuracy, filter methods can quickly obtain the final feature subset throughout the heuristic algorithms [33, 34, 37]. In addition, embedded methods have been proposed [5, 37]. It is known that feature selection in embedded methods is incorporated as a lot part of training process, which considers the empirical risk of given data. Conversely, the test process of embedded methods is dependent on the optional classifier itself.

There have been some beneficial attempts to combine feature selection with the-state-of-art learner, support vector machines (SVMs). Weston et al. proposed a wrapper method combining SVM with feature selection [32]. Owing to good generalization performance of SVM and the outstanding experimental results, similar wrapper methods have been discussed [4, 16, 30]. SVM-RFE (recursive feature elimination) is one of significant SVM-based embedded methods [15, 31]. SVM-RFE behaves well in binary classification problems. To make SVM-RFE applicable to multi-class classification problems, multi-class SVM-RFE (MSVM-RFE) methods have been proposed in [36] and [28]. Zhou and Tuck considered the linear kernel case [36], while Shieh and Yang gave the nonlinear feature selection [28]. Actually, MSVM-RFE treats a multi-class classification problem as multiple binary classification ones, where an SVM is used to solve a binary classification problem. In the linear case, the weight vectors obtained by multiple SVMs are summed as the feature weights. Then MSVM-RFE would remove the feature with the minimal weight coefficient, which is the most unimportant feature. This process is repeated until all features are ranked. In the nonlinear case, the ranking criterion considers the difference between the dual objective with all remained features and the dual objective with removing one remained feature, which leads to a situation of high computational complexity. In theory, the nonlinear MSVM-RFE has a much higher computational complexity than the linear MSVM-RFE does even if both methods adopt the linear kernel and would result in the same feature ranking.

With the purpose of settling the abnormal data detection problems, Jeong et al. applied support vector data description (SVDD) to feature selection and proposed two methods, SVDD-radius-recursive feature elimination (SVDD-RRFE) and SVDD-dual-objective-recursive feature elimination (SVDD-DRFE) [18]. Both algorithms build up a compact SVDD model to select the required features with only one-class samples. The criterion rule of SVDD-RRFE is related to the radius of an SVDD model, and that of SVDD-DRFE to the dual objective function. However, both SVDD-RRFE and SVDD-DRFE suffer from high computational complexity since both methods consider the nonlinear kernel. Pursuing the rapid feature selection in cancer classification using gene expression data, Cao et al. presented a multiple SVDD-RFE (MSVDD-RFE) method in the linear case [3]. MSVDD-RFE independently constructs multiple SVDD models and selects feature subsets according to the direction energy of model centers. A final feature subset can be obtained by merging these feature subsets. Experimental results provided in [3] show that MSVDD-RFE is much faster than both SVDD-RRFE and SVDD-DRFE. However, MSVDD-RFE could not get a final feature ranking since it generates multiple feature rankings. Thus, we do not know which feature is the most important one for the task at hand even if we have the final feature subset.

In this paper, two new fast feature selection methods based on the radius and the dual-objective ranking criteria are proposed for one-class, binary and multi-class classification problems, called fast multiple SVDD-RRFE (FMSVDD-RRFE) and fast multiple SVDD-DRFE (FMSVDD-DRFE). Compared to SVDD-RRFE and SVDD-DRFE, FMSVDD-RRFE and FMSVDD-DRFE can address not only one-class problems but also binary or multi-class classification problems. For one-class classification tasks, FMSVDD-RRFE has the same feature ranking as SVDD-RRFE, while FMSVDD-DRFE has the same result as SVDD-DRFE. However, FMSVDD-RRFE and FMSVDD-DRFE can faster rank features. For binary or multi-class classification, the proposed methods require training two or more SVDD models on which we can calculate the ranking score criteria for all features. The computational complexity of both FMSVDD-RRFE and FMSVDD-DRFE is similar to MSVDD-RFE proposed in [3].

This paper has two contributions. First, we provide speed schemes for computing radius and dual-objective ranking scores under the condition of using linear kernel, respectively. Second, based on the fast schemes, we extend the application of SVDD-RRFE and SVDD-DRFE to multi-class classification including binary classification tasks and develop FMSVDD-RRFE and FMSVDD-DRFE. The rest of the paper is organized as follows. Section 2 gives a brief presentation to related work of SVDD and SVDD-based feature selection. The new algorithms are introduced in Section 3. Section 4 discusses experimental results on

the UCI database and microarray datasets, and Section 5 provides summaries and conclusions.

2 Related work

This section discusses the related works. We simply describe SVDD and three previous SVDD-based feature selection methods [3, 18]. We assume that the data show variances in all feature directions.

2.1 Support vector data description

Inspired by SVM, Tax and Duin put forward SVDD to detect novel data or outliers [29]. SVDD can construct a closed hypersphere border surrounding the target data. Outliers are rejected outside the boundary. In other words, the hypersphere model can separate the normal (target) data and novel points. No limited to one-class problems, SVDD is also applied in dealing with multi-classification problems [24].

Let $\{\mathbf{x}_i | \mathbf{x}_i \in \mathbb{R}^D, i = 1, \ldots, n\}$ be the set of target training data, where D is the number of features and n is the number of target samples. The principle of SVDD dividing the target samples from others can be interpreted as the optimization of a convex quadratic programming:

$$\begin{aligned} &\min \quad R^2 + C \sum_{i=1}^{n} \xi_i \\ &\text{s. t. } ||\mathbf{x}_i - \mathbf{a}||^2 \leq R^2 + \xi_i, \, \xi_i \geq 0, i = 1, \cdots, n \end{aligned} \tag{1}$$

where R is the radius of the hypersphere, $\mathbf{a}$ means the center of the hypersphere, ξ_i is a slack variable, and $C > 0$ is the penalty factor which controls the balance between the volume of the model and the number of data outside the model. An unseen sample $\bar{\mathbf{x}}$ would be judged as a novel point if it satisfies

$$||\bar{\mathbf{x}} - \mathbf{a}||^2 > R^2. \tag{2}$$

By introducing the Lagrange multiplier technology, we can derive the dual programming of (1):

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{n} \alpha_i \mathbf{x}_i^T \mathbf{x}_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \mathbf{x}_i^T \mathbf{x}_j \\ &\text{s.t.} \sum_{i=1}^{n} \alpha_i = 1, 0 \leq \alpha_i \leq C, i = 1, \cdots, n, \end{aligned} \tag{3}$$

where α_i is the Lagrange multiplier. The hypersphere center $\mathbf{a}$ and radius R can be calculated with α_i, respectively. Namely, we have

$$\mathbf{a} = \sum_{i=1}^{n} \alpha_i \mathbf{x}_i \tag{4}$$

and

$$R^2(\mathbf{x}_{sv}) = \|\mathbf{x}_{sv} - \mathbf{a}\|^2 \tag{5}$$

where $\mathbf{x}_{sv}$ represents the support vector (SV) whose coefficient satisfies $0 < \alpha_{sv} < C$.

2.2 SVDD-RRFE

SVDD discriminates between normal objects and outliers for establishing a boundary (hypersphere) as compact as possible while the radius of the hypersphere refers to the compact description of the boundary. Thus, the size of radius is very important in SVDD. SVDD-RRFE considers the radius R as its ranking criterion [18].

Let SV be the set of support vectors (SVs) and J_r be the average of $R^2(\mathbf{x}_{sv})$ on all SVs. Then J_r can be defined as follows:

$$\begin{aligned} J_r &= \sum_{\mathbf{x}_{sv} \in SV} \frac{R^2(\mathbf{x}_{sv})}{|SV|} \\ &= \frac{1}{|SV|} \sum_{\mathbf{x}_{sv} \in SV} (\mathbf{x}_{sv}^T \mathbf{x}_{sv} - 2 \sum_{i=1}^{n} \alpha_i \mathbf{x}_i^T \mathbf{x}_{sv} \\ &\quad + \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \mathbf{x}_i^T \mathbf{x}_j) \end{aligned} \tag{6}$$

Let J_r^k be the average of $R^2(\mathbf{x}_{sv}^k)$, where $\mathbf{x}_{sv}^k$ means $\mathbf{x}_{sv}$ without feature k.

$$\begin{aligned} J_r^k &= \sum_{\mathbf{x}_{sv} \in SV} \frac{R^2(\mathbf{x}_{sv}^k)}{|SV|} \\ &= \frac{1}{|SV|} \sum_{\mathbf{x}_{sv} \in SV} ((\mathbf{x}_{sv}^k)^T \mathbf{x}_{sv}^k - 2 \sum_{i=1}^{n} \alpha_i (\mathbf{x}_i^k)^T \mathbf{x}_{sv}^k \\ &\quad + \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j (\mathbf{x}_i^k)^T \mathbf{x}_j^k) \end{aligned} \tag{7}$$

SVDD-RRFE aims at removing feature p which makes the radius ranking score $(J_r - J_r^p)$ minimal. Namely,

$$p = \arg \min_{k=1,\cdots,D} (J_r - J_r^k) \tag{8}$$

If J_r^k approaches to J_r, then the difference between them is small, which means the effect of the feature k is tiny. In this way, the feature with the smallest $(J_r - J_r^k)$ should be eliminated.

2.3 SVDD-DRFE

Different from the radius ranking criterion, SVDD-DRFE takes the dual objective function as the ranking criterion [18]. Let J_d be the dual objective function of SVDD. Then,

$$J_d = \sum_{i=1}^{n} \alpha_i \mathbf{x}_i^T \mathbf{x}_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \mathbf{x}_i^T \mathbf{x}_j \tag{9}$$

Let J_d^k be the dual objective function of SVDD without feature k. Then,

$$J_d^k = \sum_{i=1}^{n} \alpha_i (\mathbf{x}_i^k)^T \mathbf{x}_i^k - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j (\mathbf{x}_i^k)^T \mathbf{x}_j^k \tag{10}$$

Aiming at removing the worst feature k, SVDD-DRFE should take the smallest $(J_d - J_d^k)$ as the dual-objective ranking score. However, SVDD-DRFE proposed in [18] is to remove feature k with the largest $(J_d - J_d^k)$. In theory, if a feature is completely useless, it should not significantly change the dual objective. Thus, J_d^k should be very approached to J_d in this situation. Therefore, the worst feature p should conform with:

$$p = \arg \min_{k=1,\cdots,D} (J_d - J_d^k) \tag{11}$$

In this paper, we take (11) as the ranking criterion of SVDD-DRFE.

2.4 Multiple SVDD-RFE

MSVDD-RFE was proposed in [3], which considers the center of the hypersphere as the ranking criterion. Let the center of an SVDD model be $\mathbf{a} = [a_1, a_2, \cdots a_D]^T$. In MSVDD-RFE, $|a_i|$ indicates the average magnitude in the i-th direction with respect to the origin, and $(a_i)^2$ measures the distribution energy of data in the i-th direction. The i-th feature seems compacte when $(a_i)^2$ is small. For a multi-class classification problem, MSVDD-RFE requires training multiple hypersphere models. For each class, MSVDD-RFE trains an SVDD and ranks features. With the solution to (3) and (4), the center of the j-th SVDD model can be expressed as:

$$\mathbf{a}^j = \left[a_1^j, a_2^j, \cdots a_D^j\right]^T \tag{12}$$

where $j = 1, 2, \cdots, c$, and c is the number of classes. When processing the j-th class, MSVDD-RFE regards feature k with the smallest energy as the worst feature. Namely,

$$k = \arg \min_{p=1,\cdots,D} \left(a_p^j\right)^2. \tag{13}$$

In MSVDD-RFE, we need to determine the remained feature number in advance. For each one-class problem, MSVDD-RFE can get a remained or ranked feature subset. A final feature subset is generated by combing these remained features. However, we can not tell which is the most important in the final feature subset. In other words, the final feature subset is not ranked one.

3 Proposed methods

As mentioned before, SVDD-based feature selection methods for one-class problems take the hypersphere radius and the dual objective of SVDD as the ranking criteria, and show their good performance on some datasets [18]. However, these methods suffer from high computational complexity for considering kernel functions when computing ranking scores. MSVDD-RFE can fast select features when only the linear kernel is applied [3]. However, MSVDD-RFE could not give a final feature ranking.

To get a final feature ranking for classification tasks and improve the speed of computing ranking scores, this section proposes two fast methods based on the radius and the dual-objective criteria in the case of linear kernel, respectively.

3.1 Fast multiple SVDD-RRFE

Consider the radius ranking criterion. The size of the radius and the position of center determine the hypersphere of SVDD, and also determine the performance of SVDD. Thus, it is reasonable to make the radius as a ranking criterion. In the following, we first present a fast SVDD-RRFE (FSVDD-RRFE) method for computing radius ranking scores with the linear kernel and then propose FMSVDD-RRFE based on FSVDD-RRFE.

3.1.1 Fast SVDD-RRFE

Let the set of target training data be $\left\{\mathbf{x}_i | \mathbf{x}_i \in \mathbb{R}^D, i = 1, \ldots, n\right\}$, where D is the number of features and n is the number of target samples. We list the computational complexity of SVDD-RRFE under the situation with the linear kernel in Table 1, where D' is the number of the current feature subset and SV is the set of SVs in the current iteration. To remove the worst feature from the current feature subset, SVDD-RRFE requires calculating J_r in (6) and $J_r^k, k = 1, \cdots, D'$ in (7). The computational complexity of J_r is $O(|SV|^3 D')$, and that of J_r^k is $O(|SV|^3 (D' - 1))$. Totally, the computational complexity of feature selection is $O(|SV|^3 D'^2)$ for SVDD-RRFE in the current iteration.

Table 1 Comparison of computational complexity in an iteration

Method	Current dimensionality	Computational complexity
SVDD-RRFE	D'	$O(\|SV\|^3 D'^2)$
FSVDD-RRFE	D'	$O(\|SV\| D')$
SVDD-DRFE	D'	$O(\|SV\|^2 D'^2)$
FSVDD-DRFE	D'	$O(\|SV\|^2 D')$

To reduce the computational complexity, we propose a theorem on calculating the radius ranking score. Let the ranking score be $JR_k = J_r - J_r^k$.

Theorem 1 *Given the center* $\boldsymbol{a}$ *of the hypersphere and the set of support vectors SV obtained by SVDD with the linear kernel, the radius ranking score can be computed as*

$$JR_k = \frac{\sum_{\boldsymbol{x}_{sv}\in SV}\left(x_{(sv,k)}^2 - 2x_{(sv,k)}a_k + a_k^2\right)}{|SV|} \tag{14}$$

where $x_{(sv,k)}$ *is the k-th component of the support vector* $\boldsymbol{x}_{sv}$*, and* a_k *is the k-th component of the center* $\boldsymbol{a}$*.*

The proof of Theorem 1 is given in Appendix A. According to Theorem 1, we can construct a speeded algorithm for SVDD-RRFE. Moreover, the computational complexity of FSVDD-RRFE is $O(|SV|D')$ in an iteration, which is much smaller than that of the original algorithm SVDD-RFE. Table 1 lists the comparison of computational complexity.

3.1.2 FMSVDD-RRFE

Based on the fast algorithm proposed above, we discuss the extension of SVDD-RRFE to binary and multi-class classification.

For a c-class classification problem, assume that there is a set of training samples $X = \{\mathbf{x}_i, y_i\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^D$, $y_i \in \{1, 2, \cdots, c\}$ denotes the class label of $\mathbf{x}_i$, D and n are the number of features and samples, respectively.

Let X_j be the set of training samples in the j-th class. Then we have $X = \cup_{j=1}^c X_j$. For the j-th class, we use X_j to train an SVDD and get the corresponding center $\mathbf{a}^j$ and support vector set SV^j. Totally, there are c SVDD models. To find the worst feature by using all c SVDD models, we redesign the radius ranking score in (14) as follows:

$$JR_k = \sum_{j=1}^{c} \frac{\sum_{\mathbf{x}_{sv}\in SV^j}\left(x_{(sv,k)}^2 - 2x_{(sv,k)}a_k^j + (a_k^j)^2\right)}{|SV^j|} \tag{15}$$

If JR_p is the smallest one among $JR_k, k = 1, \cdots, D'$, then feature p should be removed from the current feature set. Thus, FMSVDD-RRFE concerns all classes when removing the worst features. Algorithm 1 displays the process of eliminating features in FMSVDD-RRFE. Roughly speaking, FMSVDD-RRFE removes the feature with the smallest ranking score in each iteration. In doing so, we can have a feature ranking at last.

Algorithm 1 FMSVDD-RRFE

Input: c-class training samples $\{\mathbf{x}_i, y_i\}_{i=1}^n$ with $\mathbf{x}_i \in \mathbb{R}^D$ and $y_i \in \{1, \cdots, c\}$.
Output: Ranked feature list R.
1. Initialize: The selected feature subset $S = \{1, \cdots, D\}$ and the ranked feature list $R = \emptyset$.
2. Repeat until $S = \emptyset$:
   (1). Let $j = 1$ and ranking scores $JR_k = 0, k \in S$;
   (2). Generate X_j which is the training set of the j-th class and only consists of features in S;
   (3). Train an SVDD with X_j to get the center $\mathbf{a}^j$ and the SVs set SV^j;
   (4). Compute the ranking scores for each feature $k \in S$:

$$JR_k \leftarrow JR_k + \frac{\sum_{\mathbf{x}_{sv}\in SV^j}\left(x_{(sv,k)}^2 - 2x_{(sv,k)}a_k^j + (a_k^j)^2\right)}{|SV^j|} \tag{16}$$

   (5). If $j < c$, let $j = j + 1$ and go to Step 2(2); Otherwise continue;
   (6). Find the feature with the smallest ranking score:

$$p = \arg\min_{k\in S} JR_k \tag{17}$$

   (7). Update the ranked feature list R by adding the index of feature p into it,

$$R \leftarrow \{p\} \cup R \tag{18}$$

   (8). Update the selected feature subset S by removing the index of feature p from it,

$$S \leftarrow S - \{p\} \tag{19}$$

3.2 Fast multiple SVDD-DRFE

Similar to the radius ranking criterion, the dual-objective ranking criterion is also reasonable since it requires maximizing the objective function to find an SVDD model. Here, we also provide a speeded algorithm for computing the dual-objective ranking scores, and then present a fast multiple SVDD-DRFE method based the speeded algorithm.

3.2.1 Fast SVDD-DRFE

The computational complexity of SVDD-DRFE with the linear kernel is also given in Table 1. Similarly, SVDD-DRFE first needs to compute J_d in (9) and $J_d^k, k = 1, \cdots, D'$ in (10), and then finds the worst feature p according to the dual-objective ranking criterion (11). The computational complexity of J_d is $O(|SV|^2 D')$, and that of J_d^k is $O(|SV|^2(D' - 1))$. Totally, the computational complexity of feature selection is $O(|SV|^2 D'^2)$ for SVDD-DRFE in the current iteration.

We also give a theorem on computing the dual-objective ranking score in the following. Let the dual-objective ranking score be $JD_k = J_d - J_d^k$.

Theorem 2 *Given the optimal solution $\alpha_i, i = 1, \cdots, n$ to the dual objective (3) in the linear case, the dual-objective ranking score can be computed as*

$$JD_k = \sum_{\boldsymbol{x}_{sv} \in SV} \alpha_{sv} x_{(sv,k)}^2 - \sum_{\boldsymbol{x}_{sv} \in SV} \sum_{\boldsymbol{x}_{sv'} \in SV} \alpha_{sv} \alpha_{sv'} x_{(sv,k)} x_{(sv',k)} \tag{20}$$

where SV is the set of support vectors, $x_{sv,k}$ is the k-th component of the support vector $\boldsymbol{x}_{sv}$, and α_{sv}, the corresponding coefficient of $\boldsymbol{x}_{sv}$, is a component of the optimal solution.

The proof of Theorem 2 is shown in Appendix B. According to Theorem 2, a fast SVDD-DRFE method can be constructed. Moreover, the computational complexity of FSVDD-DRFE is $O(|SV|^2 D')$ in an iteration, which is also greatly smaller than that of SVDD-DRFE.

Table 1 compares the computational complexity of four algorithms. Obviously, FSVDD-RRFE has the lowest complexity among four algorithms, and followed by FSVDD-DRFE. According to Table 1, SVDD-DRFE is faster than SVDD-RRFE. In other words, the speeded scheme on SVDD-RRFE is more effective than that on SVDD-DRFE, which will be proved by experiments later.

3.2.2 FMSVDD-DRFE

Based on the proposed speeded algorithm, we develop a new fast feature selection algorithm for binary and multi-class classification problems.

For a c-class classification problem, FMSVDD-DRFE also requires training c SVDD models. Let X_j be the set of training samples in the j-th class. For the j-th class, we use X_j to train an SVDD and get the solution $\alpha_i^j, i = 1, \cdots, n$. The support vector set $SV^j = \{\alpha_i^j | \alpha_i^j > 0, i = 1, \cdots, n\}$. To find the worst feature by using all c SVDD models, we rewrite the dual-objectives ranking score in (20) as follows:

$$\begin{aligned} JD_k = & \sum_{j=1}^{c} \sum_{\mathbf{x}_{sv} \in SV^j} \alpha_{sv}^j x_{(sv,k)}^2 \\ & - \sum_{j=1}^{c} \sum_{\mathbf{x}_{sv} \in SV^j} \sum_{\mathbf{x}_{sv'} \in SV^j} \alpha_{sv}^j \alpha_{sv'}^j x_{(sv,k)} x_{(sv',k)} \end{aligned} \tag{21}$$

If JD_p is the smallest one among $JD_k, k = 1, \cdots, D'$, then feature p should be removed from the current feature subset. FMSVDD-DRFE considers all classes and removes the feature with the smallest ranking score JD_k in an iteration. Algorithm 2 shows the process of feature selection in FMSVDD-DRFE which can give a feature ranking finally.

Algorithm 2 FMSVDD-DRFE

Input: c-class training samples $\{\mathbf{x}_i, y_i\}_{i=1}^n$ with $\mathbf{x}_i \in \mathbb{R}^D$ and $y_i \in \{1, \cdots, c\}$.
Output: Ranked feature list R.

1. Initialize: The selected feature subset $S = \{1, \cdots, D\}$ and the ranked feature list $R = \emptyset$.
2. Repeat until $S = \emptyset$:
   (1). Let $j = 1$ and ranking scores $JD_k = 0, k \in S$;
   (2). Generate X_j which is the training set of the j-th class and only consists of features in S;
   (3). Train an SVDD with X_j to get model coefficients $\alpha_i^j, i = 1, \cdots, n$ and the SVs set SV^j;
   (4). Compute the ranking scores for each feature $k \in S$:

$$\begin{aligned} JD_k \leftarrow \; & JD_k + \sum_{\mathbf{x}_{sv} \in SV^j} \alpha_{sv}^j x_{(sv,k)}^2 - \\ & \sum_{\mathbf{x}_{sv} \in SV^j} \sum_{\mathbf{x}_{sv'} \in SV^j} \alpha_{sv}^j \alpha_{sv'}^j x_{(sv,k)} x_{(sv',k)} \end{aligned} \tag{22}$$

   (5). If $j < c$, then $j = j + 1$ and go to Step 2(2); Otherwise continue;
   (6). Find the feature with the smallest ranking score:

$$p = \arg\min_{k \in S} JD_k \tag{23}$$

   (7). Update the ranked feature list R by adding the index of feature p into it,

$$R \leftarrow \{p\} \cup R \tag{24}$$

   (8). Update the selected feature subset S by removing the index of feature p from it,

$$S \leftarrow S - \{p\} \tag{25}$$

3.3 Connection to other SVDD-based feature selection methods

Presently, three existing SVDD-based feature selection methods, SVDD-RRFE [18], SVDD-DRFE [18], and MSVDD-RFE [3], correspond to three ranking criteria: the radius, the dual-objective and the center.

Similar to SVDD-RRFE, the proposed FMSVDD-RRFE uses the radius ranking criterion. Both FMSVDD-RRFE and SVDD-RRFE can process one-class classification problems. In this case, FMSVDD-RRFE has the same ranking result as SVDD-RRFE according to Theorem 1. But most of all, FMSVDD-RRFE is much faster than SVDD-RRFE when performing feature selection. Table 1 shows the computational complexity of the two methods when dealing with one-class classification problems.

Compared to SVDD-DRFE, the proposed FMSVDD-DRFE utilizes the same ranking criterion. When dealing with one-class tasks, FMSVDD-DRFE has the same results as and a much faster speed than SVDD-DRFE. The former can be validated by Theorem 2, and the latter can be observed from Table 1.

MSVDD-RFE uses the center ranking criterion which is different from both FM-SVDD-RRFE and FMSVDD-DRFE. The three multiple SVDD methods can address one-class, binary and multi-class tasks. MSVDD-based methods use the samples belonging to the same class to train an SVDD model. However, MSVDD-RFE considers a feature ranking only for one class instead of for all classes and generate multiple feature rankings. On the contrary, both FMSVDD-RRFE and FMSVDD-DRFE can give a unique feature ranking for all classes. In addition, the three MSVDD-based methods have a similar computational complexity which includes two parts: training c SVDD models for c-class tasks, and computing ranking scores. Obviously, the complexity of training SVDD models is exactly the same to each other. The computational complexity of computing ranking score in MSVDD-RFE is $O(|SV|D')$ in an iteration when considering one-class tasks. While the computational complexities of FMSVDD-RRFE and FMSVDD-DRFE are $O(|SV|D')$ and $O(|SV|^2D')$, respectively. Generally, SV is only a small part of training samples. When D' is large enough, $|SV|$ could be ignored.

4 Comparative performance analysis

To validate the performance of our proposed algorithms (FMSVDD-RRFE and FM-SVDD-DRFE), we compare them with other SVDD-based methods mentioned above. All numerical experiments are performed on a personal computer with a 3.4GHz Intel Core and 4G bytes of memory. This computer runs Windows 7, with Matlab R2013a.

4.1 Simulated datasets

The goal of this experiment is to validate that the speeded algorithms can improve the speed of feature ranking. We generate two simulated datasets, Dataset A and Dataset B. Dataset A only contains one-class data, and Dataset B consists of two-class data. In the following, we describe the experimental results on the two datasets, respectively.

4.1.1 Dataset A

We use Dataset A to validate that the proposed speeded algorithms have the same feature ranking and a faster ranking speed compared to the original ones when applying these algorithm to one-class tasks. For Dataset A, we first randomly generate two-dimensional data with a uniform distribution on the interval [0, 1] and then add noise features which are m-Gaussian distributions with zero mean and a variance 0.01, where m takes value in the set $\{2^1, 2^2, \cdots, 2^{12}\}$. In other words, the feature number in Dataset A is $m + 2$. The number of training samples is 50.

The compared methods are FMSVDD-RRFE, FMSVDD-DRFE, SVDD-RRFE, and SVDD-DRFE. Actually, FMSVDD-RRFE is FSVDD-RRFE, and FMSVDD-DRFE is FSVDD-DRFE since only one SVDD model is needed. Let $C = 0.5$ for all four methods here. We perform 10 runs for each m, and report the average feature ranking time in Fig. 1. Note that the logarithmic base 10 scale is used for the Y-axis. Naturally, all four methods spend much time ranking feature with the increase of feature number. The curves in Fig. 1 lead to the same conclusion as Table 1 shows. When the feature number is much larger than the sample number, SVDD-RRFE has the highest computational complexity, followed by SVDD-DRFE. For example, when $m = 2^{10}$, the average ranking time of SVDD-RRFE is 637.75 second, SVDD-DRFE 432.27 second, FMSVDD-DRFE 8.11 second, and FMSVDD-RRFE 3.58 second. In this case, the proposed methods are two orders of magnitude faster than the old ones.

Now, we observe the ranked features. Without loss of generality, let $m = 2^4$. Then, Dataset A has 18 features where the first two features are useful and the rest ones are noise. By randomly generating data, we perform 10 trials. The ranked feature indices in one trial are listed in Table 2. Experimental results of the other nine trials are similar to the one listed in Table 2. From the ranking sequences obtained by four methods, the first two features are correctly ranked in front of the sequence. In addition, as we expected that FMSVDD-RRFE has the same ranking as SVDD-RRFE, and FMSVDD-DRFE as SVDD-DRFE. In order words, our algorithms speed both SVDD-RRFE and SVDD-DRFE without changing their performance.

4.1.2 Dataset B

We use Dataset B to validate that the proposed methods could get better feature ranking for binary (or

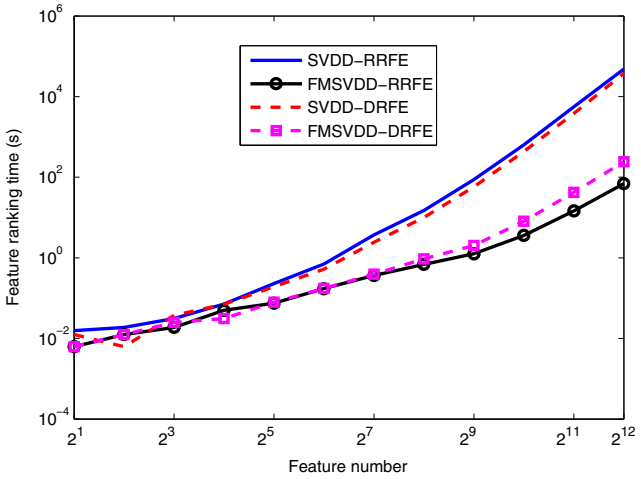


Fig. 1 Feature ranking time vs. feature number on Dataset A

Table 2 Feature ranking of different methods on Dataset A

Method	Feature ranking
SVDD-RRFE	1,2,8,16,5,3,13,14,15,6,7,18,10,9,12,11,4,17
FMSVDD-RRFE	1,2,8,16,5,3,13,14,15,6,7,18,10,9,12,11,4,17
SVDD-DRFE	1,2,8,16,3,5,13,15,14,7,6,18,4,11,10,17,9,12
FMSVDD-DRFE	1,2,8,16,3,5,13,15,14,7,6,18,4,11,10,17,9,12

multiple-class) classification problem than the original ones do. For Dataset B, we generate two-class samples in the 12-dimensional data space, the class centers of which are located at $[0, \cdots, 0]^T$ (class one) and $[1, \cdots, 1]^T$ (class two), respectively. Similar dataset was tested in [16, 18]. The ith feature of all samples is independently drawn from the Gaussian distribution with the standard deviation $0.2 \times 1.2^{i-1}$. For each class, 250 samples are randomly generated for training and 250 ones for test. The compared methods are FMSVDD-RRFE, FMSVDD-DRFE, SVDD-RRFE, and SVDD-DRFE. Let $C = 0.5$ for all four methods here. We treat class one as the normal data when applying SVDD-RRFE and SVDD-DRFE. Totally, there are 500 training samples for FMSVDD-RRFE and FMSVDD-DRFE. For all four methods, the number of test samples is 500.

In one trial, feature rankings obtained by four methods are listed in Table 3. In fact, the smaller the feature index is, the more important the feature in Dataset B is. From Table 3, we can see that feature six is ranked in the third place by FMSVDD-DRFE, and in the sixth place by SVDD-DRFE. Similarly, feature five is ranked in the fifth place by FMSVDD-RRFE, and in the sixth place by SVDD-RRFE. In other words, our methods could provide better feature ranking and result better classification performance.

To compare the classification performance of these four methods, we take support vector machine (SVM) with the linear kernel as the classifier and use the F1-measure to measure classification performance. In SVM, let the regularized parameter be 10, which is an empirical value. The F1-measure is a statistic that can evaluate the accuracy of model by combining precision and recall:

$$F1 = \frac{1}{c}\sum_{j=1}^{c}\frac{2 \times Precision_j \times Recall_j}{Precision_j + Recall_j} \tag{26}$$

Table 3 Feature ranking of different methods on Dataset B

Method	Feature ranking
SVDD-RRFE	12,11,10,7,9,5,4,8,6,3,2,1
FMSVDD-RRFE	12,11,9,10,5,7,4,6,8,1,3,2
SVDD-DRFE	12,11,10,7,9,6,8,5,2,4,1,3
FMSVDD-DRFE	12,11,6,9,5,10,4,7,1,8,3,2

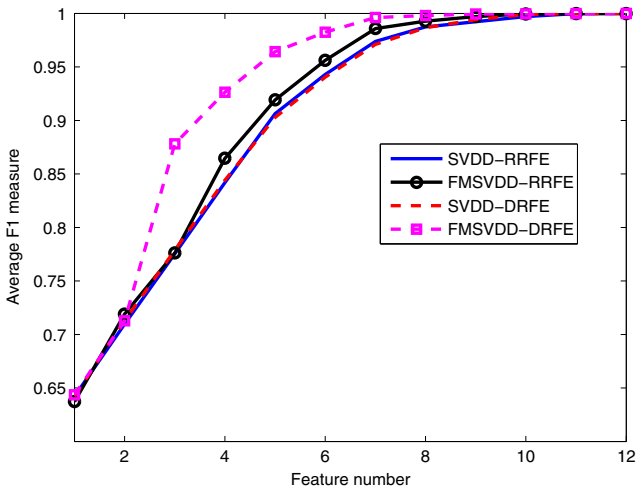


Fig. 2 Average F1 measure vs. feature number on Dataset B

where $Precision_j = \frac{TP_j}{TP_j+FP_j}$ and $Recall_j = \frac{TP_j}{TP_j+FN_j}$, FP_j is the number of misclassified jth samples, TP_j is the number of true jth ones, and FN_j is the number of misclassified non-jth samples.

The average classification performance on 10 runs obtained by SVM is shown in Fig. 2. Obviously, FMSVDD-DRFE is always the best, followed by FMSVDD-RRFE. These results lead to a conclusion that FMSVDD-RRFE and FMSVDD-DRFE are more effective than SVDD-RRFE and SVDD-DRFE on Dataset B, which is consistent with the results listed in Table 3.

4.2 Microarray datasets

Microarray datasets describe the differential expression genes and have extensive and thorough gene expressive quantity [11], which would be accompanied by terrible computation. Feature selection methods seek the appropriate features to solve the above conflict effectively. Four public available microarray datasets are used to validate the performance of our proposed method, and summarized in Table 4 including Small Round Blue Cell Tumor (SRBCT) of Khan et al. [19], Leukemia-ALLAML of Golub et al. [13], Central Nervous System (CNS) dataset of Pomeroy et al. [26], and Lung Cancer of Bhattacharjee et al. [1].

Table 4 Description of three datasets

Dataset	# Class	# Feature	#Training sample	#Test Sample
SRBCT	4	2308	63	20
Leukemia-ALLAML	2	7129	38	34
CNS	2	7129	40	20
Lung Cancer	5	12600	54	149

In these datasets, all genes are expressed as numerical values at different measurement levels. Since the value is related to the value of genes, we normalize each gene on the interval [0, 1] so that all genes can be measured on the same scale.

Four classification performance indices are used here, including F1-measure (26), recall, precision, and accuracy, which are respectively defined by

$$Recall = \frac{1}{c}\sum_{j=1}^{c} Recall_j \tag{27}$$

$$Precision = \frac{1}{c}\sum_{j=1}^{c} Precision_j \tag{28}$$

and

$$Accuracy = \frac{1}{n'}\sum_{j=1}^{c} TP_j \tag{29}$$

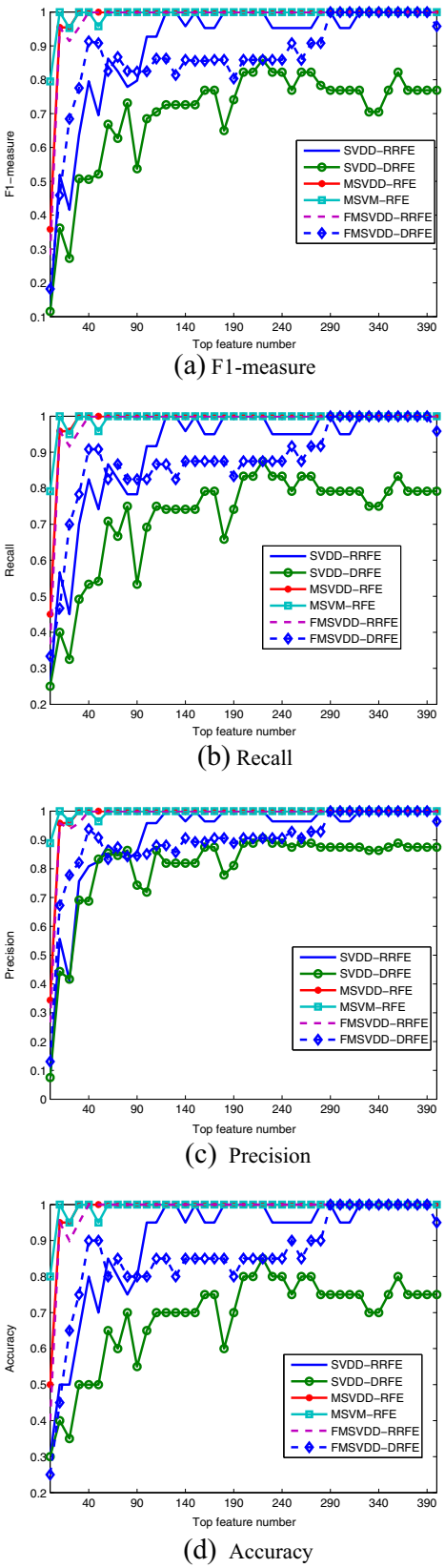
where n' is the number of test samples.

4.2.1 SRBCT

The SRBCT dataset contains 83 samples belonging to four classes, or Ewing family of tumors (EWS), neuroblastoma (NB), Burkitt lymphoma (BL) and rhabdomyosarcoma (RMS). Each sample has 2308 genes. The training set has 63 samples, including 23 EWS, 20 RMS, 12 NB and 8 BL. The test set contains 6 EWS, 6 RMS, 6NB and 3 BL. This dataset could be downloaded from http://www.biolab.si/supp/bi-cancer/projections/info/SRBCT.htm.

We compare our methods with SVDD-RRFE [18], SVDD-DRFE [18], MSVM-RFE [36], and MSVDD-RFE [3] on this dataset. In these feature selection methods, the parameter C has a vital influence on experimental results. Here, we use 5-fold cross validation to select the parameter C. The parameter C for all SVDD-based methods take values in the set $\{0.2, 0.5, 1\}$, for SVM-based methods take values in the set $\{1, 10, 100\}$. Both SVDD-RRFE and SVDD-DRFE require only one-class samples. In the SRBCT dataset, there are 23 training samples in class EWS which is taken as the target class in our experiments.

After we get the feature ranking list, we deal with the top 400 features in the list. We show the classification performance from the top 1 to 400 features on the SRBCT dataset in Fig. 3, where SVM with the linear kernel is the subsequent classifier. When the feature number is small (say less than 40), MSVM-RFE, MSVDD-RFE and FMSVDD-RRFE have a rather good performance compared to the rest methods. SVDD-DRFE performs worst among these six methods. Since FMSVDD-DRFF is proposed based on SVDD-DRFE, FMSVDD-DRFE is not so good. Although



(a) F1-measure

(b) Recall

(c) Precision

(d) Accuracy

Fig. 3 Comparison of classification performance (**a**) F1-measure (**b**) Recall, (**c**) Precision (**d**) Accuracy vs. top feature number on the SRBCT datasets

Table 5 Performance comparison on SRBCT

Performance	SVDD-RRFE	SVDD-DRFE	MSVDD-RFE	MSVM-RFE	FMSVDD-RRFE	FMSVDD-DRFE
Feature number	114	202	92	3	11	282
F1-measure (%)	100.00	85.90	100.00	100.00	100.00	100.00
Recall (%)	100.00	87.50	100.00	100.00	100.00	100.00
Precision (%)	100.00	90.62	100.00	100.00	100.00	100.00
Accuracy (%)	100.00	85.00	100.00	100.00	100.00	100.00
Ranking time (sec.)	761.24	647.72	19.50	86.44	22.56	20.19

FMSVDD-DRFE is bad, it is still much better than SVDD-DRFE. Similarly, FMSVDD-RRFF is proposed based on SVDD-RRFE and much better than SVDD-RRFE.

The best performance obtained by these methods is given in Table 5, where feature number is determined by the values of best F1-measure of these methods. The other three performance indices recall, precision and accuracy are determined according to the selected feature number. All methods achieve good classification performance except for SVDD-DRFE. FMSVDD-RRFE requires less genes to implement perfect classification than SVDD-RRFE.

The feature ranking time on the training set is also listed in Table 5. We can see that MSVDD-RFE, FMSVDD-RRFE and FMSVDD-DRFE have a comparable ranking time, which supports the analysis in Section 3.3, or the three MSVDD-RFE methods have similar computational complexity. SVM-RFE is slow by comparison. In addition, both SVDD-RRFE and SVDD-DRFE are much slower than other methods.

4.2.2 Leukemia-ALLAML

The Leukemia-ALLAML dataset has 72 samples belonging to two classes, or ALL (Acute Lymphoblastic Leukemia) and AML (Acute Myeloid Leukemia). The training set consists of 38 bone marrow samples (27 ALL and 11 AML), over 7129 probes from 6817 human genes. In addition, 34 test samples is provided, with 20 ALL and 14 AML. This dataset could be downloaded from http://datam.i2r.a-star.edu.sg/datasets/krbd/Leukemia/ALLAML.html.

Since both SVDD-RRFE and SVDD-DRFE took a long time to rank feature on the Leukemia-ALLAML dataset, we replace them by FSVDD-RRFE and FSVDD-DRFE, respectively. In theory, FSVDD-RRFE has the same feature list as SVDD-RRFE, and FSVDD-DRFE as SVDD-DRFE. But FSVDD-RRFE and FSVDD-DRFE are much faster than SVDD-RRFE and SVDD-DRFE, respectively. For FSVDD-RRFE and FSVDD-DRFE, 27 ALL are taken as the target samples. Parameter setting is the same as that in the SRBCT dataset.

We report the best performance obtained by six methods in Table 6. These results lead to similar conclusions as those on the SRBCT dataset. FMSVDD-RRFE can achieve the best classification performance as well as MSVDD-RFE and MSVM-RFE. FMSVDD-RRFE is much better than FSVDD-RRFE, and FMSVDD-DRFE better than FSVDD-DRFE. Note that FSVDD-RRFE and FSVDD-DRFE have a faster ranking speed since the two methods deal with only one-class samples.

4.2.3 More datasets

The CNS dataset contains 60 patient samples, 21 are survivors (alive after treatment) and 39 are failures (succumbed to their disease). There are 7129 genes in the dataset. The training set consists of the first 10 survivors and 30 failures, the other 11 survivors and 9 failures are testing points. The CNS dataset could be downloaded from http://datam.i2r.a-star.edu.sg/datasets/krbd/NervousSystem/NervousSystem.html.

Table 6 Performance comparison on Leukemia-ALLAML

Performance	FSVDD-RRFE	FSVDD-DRFE	MSVDD-RFE	MSVM-RFE	FMSVDD-RRFE	FMSVDD-DRFE
Feature number	30	30	28	145	10	8
F1-measure (%)	96.93	79.84	100.00	100.00	100.00	93.93
Recall (%)	96.43	78.57	100.00	100.00	100.00	93.93
Precision (%)	97.62	88.46	100.00	100.00	100.00	93.93
Accuracy (%)	97.06	82.35	100.00	100.00	100.00	94.12
Ranking time (sec.)	171.74	214.69	346.04	554.18	339.64	346.29

This Lung Cancer dataset has a total of 203 snap-frozen lung tumors and normal lung. The 203 speciments include 139 samples of lung adenocarcinomas (labeled as ADEN), 21 samples of squamous cell lung carcinomas (labeled as SQUA), 20 samples of pulmonary carcinoids (labeled as COID), 6 samples of small-cell lung carcinomas (labeled as SCLC) and 17 normal lung samples (labeled as NORMAL). Each sample is described by 12600 genes. The first 20 ADEN, 10 SQUA, 10 COID, 4 SCLC, and 10 NORMAL are taken as the training samples, and the rest are the test ones. This dataset could be downloaded from http://datam.i2r.a-star.edu.sg/datasets/krbd/LungCancer/LungCancer-Harvard1.html.

For the CNS dataset, 30 failures are the target samples in FSVDD-RRFE and FSVDD-DRFE. For the Lung Cancer dataset, ADEN is selected as the target class. Since four performance indices F1-measure, recall, precision and accuracy are coincident according to Tables 5 and 6, we only give the best F1-measure performance in Table 7. First, we have a conclusion that FMSVDD-RRFE is equal to or better than FSVDD-RRFE, and FMSVDD-DRFE is always better than FSVDD-DRFE. On the CNS dataset, although FSVDD-DRFE is the worst, FMSVDD-DRFE achieves the best performance 79.80% among six methods. FMSVDD-RRFE, FSVDD-RRFE and MSVDD-RFE are next to FMSVDD-DRFE. On the Lung Cancer dataset, MSVDD-RFE achieves the best 95.18%, followed by FMSVDD-RRFE.

In a nutshell, FMSVDD-RRFE and FMSVDD-DRFE can get not only a better classification performance but also a faster feature ranking speed than SVDD-RRFE and SVDD-DRFE on four microarray datasets, respectively. FMSVDD-RRFE has a comparable classification performance with MSVM-RFE and MSVDD-RFE on four datasets. FMSVDD-DRFE only behaves well on the CNS dataset.

4.3 UCI database

This section considers the datasets where the sample number is much greater than the feature number. We perform experiments on eight datasets from the UCI database [9], Breast, Wine, Wdbc, Vowel, Vehicle, Soy, Waveform, and Segment datasets. The description on these eight datasets is shown in Table 8. These datasets are normalized so that their features range in the interval [0, 1]. For each dataset, we randomly split it into two subsets for training and test, respectively. The training set contains 2/3 of the samples of each class, and the test set contains the remaining 1/3. The split process is repeated 10 times for each dataset.

We also compare the proposed FMSVDD methods with other four feature selection methods, SVDD-RRFE, SVDD-DRFE, MSVDD-RFE and MSVM-RFE. The parameter C for all SVDD-based methods take values in the set $\{0.1, 0.5, 1\}$, for SVM-based methods take values in the set $\{1, 10, 100\}$. We use 5-fold cross validation to select the parameter C and the optimal feature number on the training set for all compared methods. Note that, for both SVDD-RRFE and SVDD-DRFE, only one-class samples are supported. We choose the class with the largest number of samples, or randomly choose the target class when each class has the same number of samples.

We report the average classification performance of these methods on 10 test subsets in Tables 9, 10, 11 and 12, where SVM with the linear kernel is the subsequent classifier and the bold values are the best ones among the compared methods. From Tables 9-12, we can see that these four performance indices are consistent with each other, which also shows that the performance of algorithms is stable even for imbalanced data. For example, the ratio of sample numbers between two classes is almost 2 : 1 in the Breast dataset. It is obvious that MFSVDD-RRFE achieves the best F1-measure in five out of eight datasets. FMSVDD-RRFE outperforms both SVDD-RREF and MSVM-RFE in seven out of eight data sets, and both SVDD-DRFE and MSVDD-RFE in all eight datasets. In addition, FMSVDD-DRFE is better than both SVDD-RRFE and MSVDD-RFE in seven out of eight datasets, SVDD-DRFE in all eight datasets, and MSVM-RFE in five out of eight datasets. FMSVDD-DRFE only outperforms FMSVDD-RRFE on the Vehicle dataset.

In summary, FMSVDD-RRFE and FMSVDD-DRFE are always better than SVDD-RRFE and SVDD-DRFE when addressing the binary or multi-class tasks where the sample

Table 7 Comparison of F1-measure (%) obtained by six methods

Method	CNS	Lung Cancer
FSVDD-RRFE	70.00	88.70
FSVDD-DRFE	64.91	84.65
MSVDD-RFE	70.00	92.62
MSVM-RFE	64.91	95.18
FMSVDD-RRFE	70.00	93.40
FMSVDD-DRFE	79.80	91.11

Table 8 Description on eight UCI Datasets

Dataset	# Attribute	# Sample	# Class
Breast	9	699	2
Wdbc	569	30	2
Wine	13	178	3
Vowel	10	660	12
Vehicle	18	846	4
Soy	208	289	17
Waveform	21	5000	3
Segment	19	2310	7

Table 9 Average accuracy of eight UCI Datasets

Dataset	SVDD-RRFE	SVDD-DRFE	MSVDD-RFE	MSVM-RFE	FMSVDD-RRFE	FMSVDD-DRFE
Breast	96.36±0.75	96.36± 1.14	96.41±0.86	96.26±0.78	**96.65**± 0.60	96.46± 0.98
Wdbc	97.94±1.23	97.82± 1.28	97.88± 1.37	97.30±1.28	**98.06**±1.21	98.00±1.44
Wine	95.34±3.05	95.86±2.18	95.86± 3.65	96.55±3.63	**97.24**±2.18	96.20± 2.41
Vowel	**79.23**± 2.09	79.09±2.32	76.80± 1.38	79.09±2.32	79.19±2.06	79.19± 2.05
Vehicle	77.62 ± 2.46	77.66 ±2.31	77.30 ± 2.50	76.83 ± 2.91	77.70 ±2.17	**77.82**±2.73
Soy	95.69± 2.82	95.03± 2.75	95.16± 3.03	95.69±2.50	**95.95**± 2.57	95.42± 2.85
Waveform	86.71± 0.64	86.76 ± 0.65	86.82± 0.51	86.70±0.55	**86.90** ± 0.61	86.86±0.63
Segment	93.93± 0.81	93.90 ± 0.75	94.01 ± 0.67	**94.17** ± 0.70	94.05±0.87	94.01±0.72

Table 10 Average recall of eight UCI Datasets

Dataset	SVDD-RRFE	SVDD-DRFE	MSVDD-RFE	MSVM-RFE	FMSVDD-RRFE	FMSVDD-DRFE
Breast	96.14±1.01	95.97± 1.37	96.11±1.22	95.96±1.09	**96.42**± 0.86	96.18± 1.16
Wdbc	97.68±1.59	97.55± 1.62	97.63± 1.76	96.84±1.51	**97.77**±1.50	97.69±1.70
Wine	95.68±2.80	96.30±1.89	96.34± 3.46	96.79±3.43	**97.49**±1.98	96.59± 2.25
Vowel	**79.22**± 2.09	79.09±2.32	76.80± 1.38	79.09±2.32	79.19±2.06	79.19± 2.05
Vehicle	77.96± 2.42	78.00±2.28	77.65± 2.46	77.18± 2.86	78.04 ±2.15	**78.16**±2.59
Soy	95.69± 2.82	95.03± 2.75	95.16± 3.03	95.69±2.50	**95.95**± 2.57	95.42± 2.85
Waveform	86.68± 0.64	86.74 ± 0.65	86.79± 0.51	86.67±0.55	**86.88** ± 0.62	86.84±0.63
Segment	93.93± 0.81	93.90 ± 0.75	94.01 ± 0.67	**94.17** ± 0.70	94.05±0.87	94.01±0.72

Table 11 Average precision of eight UCI Datasets

Dataset	SVDD-RRFE	SVDD-DRFE	MSVDD-RFE	MSVM-RFE	FMSVDD-RRFE	FMSVDD-DRFE
Breast	95.87±0.84	96.01± 1.34	96.00±0.91	95.83±0.88	**96.21**± 0.70	96.04± 1.21
Wdbc	97.93±1.11	97.79± 1.19	97.84± 1.22	97.36±1.27	**98.08**±1.16	98.03±1.45
Wine	95.44±2.81	95.98±1.97	95.98± 3.26	96.80±3.41	**97.23**±2.08	96.26± 2.13
Vowel	**79.72**± 1.80	79.58±2.01	77.25± 1.20	79.58±2.01	79.69±1.77	79.69± 1.77
Vehicle	77.11± 2.62	77.15±2.46	76.75± 2.59	76.12± 3.07	77.20 ±2.38	**77.36**±2.81
Soy	96.31± 2.44	95.79± 2.35	95.88± 2.67	96.42±2.08	**96.58**± 2.18	96.15± 2.43
Waveform	86.73± 0.63	86.78 ± 0.65	86.83± 0.51	86.71±0.55	**86.93** ± 0.61	86.88±0.62
Segment	93.97± 0.77	93.93 ± 0.73	94.05± 0.65	**94.26** ± 0.72	94.11±0.89	94.06±0.73

Table 12 Average F1-measure values of eight UCI Datasets

Dataset	SVDD-RRFE	SVDD-DRFE	MSVDD-RFE	MSVM-RFE	FMSVDD-RRFE	FMSVDD-DRFE
Breast	95.98±0.83	95.97± 1.27	96.03±0.97	95.87±0.87	**96.30**± 0.67	96.09± 1.09
Wdbc	97.78±1.34	97.66± 1.40	97.72± 1.51	97.08±1.39	**97.91**±1.32	97.85±1.57
Wine	95.47±2.82	95.98±2.02	96.06± 3.38	96.71±3.49	**97.26**±2.10	96.34± 2.23
Vowel	**79.06**± 2.05	78.96±2.23	76.53 ± 1.40	78.96 ±2.23	79.04±2.03	79.04± 2.03
Vehicle	77.30 ± 2.64	77.32 ±2.48	76.95 ± 2.64	76.34 ± 3.07	77.38 ±2.35	**77.54**±2.82
Soy	95.56± 2.99	94.88± 2.91	95.04± 3.218	95.54 ±2.69	**95.83**± 2.73	95.25± 3.04
Waveform	86.67± 0.65	86.72 ± 0.66	86.78 ± 0.52	86.66 ±0.56	**86.86** ± 0.62	86.82±0.63
Segment	93.89 ± 0.833	93.87 ± 0.78	94.00 ± 0.66	**94.16** ± 0.77	94.04±0.93	93.99±0.78

Table 13 The mean rank of six methods on 12 datasets

SVDD-RRFE	SVDD-DRFE	MSVDD-RFE	MSVM-RFE	FMSVDD-RRFE	FMSVDD-DRFE
3.7500	5.0833	3.6667	3.7500	1.7917	2.7917

Table 14 Friedman tests with the corresponding post-hoc tests

$CD_{0.10}$	SVDD-RRFE	SVDD-DRFE	MSVDD-RFE	MSVM-RFE	FMSVDD-RRFE	FMSVDD-DRFE
FMSVDD-RRFE	1.9583	3.2917	1.8750	1.9583	0	1.0000
FMSVDD-DRFE	0.9583	2.2917	0.8750	0.9583	−1.0000	0

number is much greater than the feature number, respectively. The main reason is that proposed methods incorporate the information from all classes instead of one class. Compared to methods utilizing all class information (both MSVDD-RFE and MSVM-RFE), FMSVDD-RRFE and FMSVDD-DRFE also have superiority.

4.4 Statistical comparison over multiple datasets

In this subsection, we perform statistical tests on multiple data sets for comparing different algorithms. In the following, we conduct the Friedman test with the corresponding post-hoc tests, which is a non-parametric equivalence of the repeated-measures analysis of variance (ANOVA) under the null hypothesis that all the algorithms are equivalent and so their ranks should be equal [10]. The Friedman test is carried out to test whether all the algorithms are equivalent. If the test result rejects the null hypothesis, i.e., these algorithms are equivalent, we can proceed to a post-hoc test, or the Bonferroni-Dunn test [8]. The performance of pairwise classifiers is significantly different if the corresponding average ranks differ by at least the critical difference

$$CD = q_\alpha \sqrt{\frac{j(j+1)}{6T}} \tag{30}$$

where j is the number of algorithms, T is the number of data sets, the critical values q_α can be found in [10], and the subscript α is the threshold value. Generally, let $\alpha = 0.1$ and $q_{0.10} = 2.326$ [7]. In detail, we have $j = 6$ and $T = 12$ (including four microarray and eight UCI datasets), then $CD = 1.7765$.

Table 13 lists the mean rank of six feature selection algorithms, SVDD-RRFE, SVDD-DRFE, MSVDD-RFE, MSVM-RFE, FMSVDD-RRFE and FMSVDD-DRFE. We can see that FMSVDD-RRFE lists the top, followed by FMSVDD-DRFE. Table 14 shows the Friedman test results. According to the results in Table 14, we find that the differences between FMSVDD-RRFE and other algorithms are greater than the critical difference 1.7765 except FMSVDD-DRFE. Thus, FMSVDD-RRFE is significantly better than the other four methods. Similarly, FMSVDD-DRFE is just significantly better than SVDD-DRFE.

5 Summaries and conclusions

Recent developments of feature selection have achieved outstanding simulated results along with favorable time complexity. Promotion in the SVDD-based methods brings with it some meaningful study. This paper develops two new fast feature selection methods based on multiple support vector data description, called FMSVDD-RRFE and FMSVDD-DRFE. The proposed methods can address not only one-class classification tasks, but also binary and multi-class ones. When dealing with one-class problems and using the linear kernel, FMSVDD-RRFE is a fast version of SVDD-RRFE and FMSVDD-DRFE is a speeded version of SVDD-DRFE. The facts are proved by Theorem 1 and Theorem 2, respectively. Extensive experiments are performed to validate the performance of the proposed methods. On eight UCI datasets, both FMSVDD-RRFE and FMSVDD-DRFE behave well. On four microarray datasets, the performance of FMSVDD-RRFE is compared to that of the state-of-the-art feature selection methods, MSVM-RFE and MSVDD-RFE. More importantly, the feature ranking time of SVDD-RRFE and SVDD-DRFE is greatly reduced by our proposed methods when addressing high-dimensional datasets, which is supported by the experimental results of the simulated Dataset A and the SRBCT dataset.

The proposed methods are improved versions of SVDD-based methods with the linear kernel. Thus, our proposed methods are used only with the linear kernel, which means that they could not perform nonlinear feature selection. In the future, we will consider the improvement of nonlinear kernels, such as radius basis function (RBF) kernel.

Acknowledgments This work was supported in part by the National Natural Science Foundation of China under Grant No. 61373093, by the Natural Science Foundation of Jiangsu Province of China under Grant No. BK20140008, and by the Soochow Scholar Project.

Appendix

A. Proof of Theorem 1

Proof Assume that a linear SVDD model has been trained. Then we can get the center **a** of the hypersphere, and the set

of support vectors SV. Substituting J_r (6) and J_r^k (7) into the radius ranking score JR_k, we have

$$\begin{aligned} JR_k &= J_r - J_r^k \\ &= \sum_{\mathbf{x}_{sv}\in SV} \frac{R^2(\mathbf{x}_{sv})}{|SV|} - \sum_{\mathbf{x}_{sv}\in SV} \frac{R^2\left(\mathbf{x}_{sv}^k\right)}{|SV|} \end{aligned} \tag{31}$$

where $\mathbf{x}_{sv} \in \mathbb{R}^D$ is a support vector, and $\mathbf{x}_{sv}^k = [x_{(sv,1)}, \cdots, x_{(sv,k-1)}, x_{(sv,k+1)}, \cdots, x_{(sv,D)}]^T \in \mathbb{R}^{D-1}$.

According to (31), it is necessary to find the difference $R^2(\mathbf{x}_{sv}) - R^2(\mathbf{x}_{sv}^k)$. Let $\mathbf{a}^k = [a_1, \cdots, a_{k-1}, a_{k+1}, \cdots, a_D] \in \mathbb{R}^{D-1}$. Then, we have

$$\begin{aligned} &R^2(\mathbf{x}_{sv}) - R^2(\mathbf{x}_{sv}^k) \\ &= \|\mathbf{x}_{sv} - \mathbf{a}\|^2 - \|\mathbf{x}_{sv}^k - \mathbf{a}^k\|^2 \\ &= \mathbf{x}_{sv}^T\mathbf{x}_{sv} - 2\mathbf{x}_{sv}^T\mathbf{a} + \mathbf{a}^T\mathbf{a} - \\ &\quad \left((\mathbf{x}_{sv}^k)^T\mathbf{x}_{sv}^k - 2(\mathbf{x}_{sv}^k)^T\mathbf{a}^k + (\mathbf{a}^k)^T\mathbf{a}^k\right) \end{aligned} \tag{32}$$

Since

$$\mathbf{x}_{sv}^T\mathbf{x}_{sv} - (\mathbf{x}_{sv}^k)^T\mathbf{x}_{sv}^k = x_{(sv,k)}^2 \tag{33}$$

$$\mathbf{x}_{sv}^T\mathbf{a} - (\mathbf{x}_{sv}^k)^T\mathbf{a}^k = x_{(sv,k)}a_k \tag{34}$$

and

$$\mathbf{a}^T\mathbf{a} - (\mathbf{a}^k)^T\mathbf{a}^k = a_k^2 \tag{35}$$

we substitute (33), (34) and (35) into (32), and get

$$R^2(\mathbf{x}_{sv}) - R^2(\mathbf{x}_{sv}^k) = x_{(sv,k)}^2 - 2x_{(sv,k)}a_k + a_k^2 \tag{36}$$

Then, substituting (36) into (31), the radius ranking score can be rewritten as:

$$JR_k = \frac{1}{|SV|}\sum_{\mathbf{x}_{sv}\in SV}\left(x_{(sv,k)}^2 - 2x_{(sv,k)}a_k + a_k^2\right) \tag{37}$$

This completes the proof of Theorem 1. □

B. Proof of Theorem 2

Proof Assume that a linear SVDD model has been trained. Then we can get the coefficients $\alpha_i, i = 1, \cdots, n$ of the hypersphere, and the set of support vectors $SV = \{\alpha_i | \alpha_i > 0, i = 1, \cdots, n\}$. Substituting J_d (9) and J_d^k (10) into the dual-objective ranking score JD_k, we have

$$\begin{aligned} JD_k &= J_d - J_d^k \\ &= \sum_{i=1}^n \alpha_i \mathbf{x}_i^T\mathbf{x}_i - \sum_{i=1}^n\sum_{j=1}^n \alpha_i\alpha_j\mathbf{x}_i^T\mathbf{x}_j - \\ &\quad \sum_{i=1}^n \alpha_i(\mathbf{x}_i^k)^T\mathbf{x}_i^k - \sum_{i=1}^n\sum_{j=1}^n \alpha_i\alpha_j(\mathbf{x}_i^k)^T\mathbf{x}_j^k \end{aligned} \tag{38}$$

Since

$$\mathbf{x}_i^T\mathbf{x}_i - (\mathbf{x}_i^k)^T\mathbf{x}_i^k = x_{(i,k)}^2 \tag{39}$$

and

$$\mathbf{x}_i^T\mathbf{x}_j - (\mathbf{x}_i^k)^T\mathbf{x}_j^k = x_{(i,k)}x_{(j,k)} \tag{40}$$

we substitute (39) and (40) into (38), and get

$$JD_k = \sum_{i=1}^n \alpha_i x_{(i,k)}^2 - \sum_{i=1}^n\sum_{j=1}^n \alpha_i\alpha_j x_{(i,k)}x_{(j,k)} \tag{41}$$

Since only support vectors contribute to computing (41), we rewrite (41) as follows:

$$\begin{aligned} JD_k = &\sum_{\mathbf{x}_{sv}\in SV} \alpha_{sv}x_{(sv,k)}^2 - \\ &\sum_{\mathbf{x}_{sv}\in SV}\sum_{\mathbf{x}_{sv'}\in SV} \alpha_{sv}\alpha_{sv'}x_{(sv,k)}x_{(sv',k)} \end{aligned} \tag{42}$$

This completes the proof of Theorem 2. □

References

1. Bhattacharjee A, Richards WG, Staunton J, Li C, Monti S, Vasa P, Ladd C, Beheshti J, Bueno R, Gillette M (2001) Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses. Proc Nat Acad Sci 98(24):13,790–13,795
2. Blum A, Langley P (1997) Selection of relevant features and examples in machine learning. Artif Intell 97(1-2):245–271
3. Cao J, Zhang L, Wang B, Li F, Yang J (2015) A fast gene selection method for multi-cancer classification using multiple support vector data description. J Biomed Inf 53:381–389
4. Chen H, Yang B, Liu J, Liu D (2011) A support vector machine classifier with rough set-based feature selection for breast cancer diagnosis. Expert Syst Appl 38(7):9014–9022
5. Daelemans W, Goethals B, Morik K (eds) (2008) Machine learning and knowledge discovery in databases, european conference, ECML/PKDD 2008. In: Proceedings, part II, lecture notes in computer science, vol 5212. Springer, Antwerp
6. Dash M, Liu H (1997) Feature selection for classification. Intell Data Anal 1(1):131–156
7. Demśar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
8. Dunn OJ (1961) Multiple comparisons among means. J Amer Stat Assoc 56(293):52–64
9. Frank A, Asuncion A (2010) UCI machine learning repository from http://archive.ics.uci.edu/ml.html
10. Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Amer Stat Assoc 32(200):675–701
11. Geller SC, Gregg JP, Hagerman P, Rocke DM (2003) Transformation and normalization of oligonucleotide microarray data. Bioinformatics 19(14):1817–1823
12. Gheyas IA, Smith LS (2010) Feature subset selection in large dimensionality domains. Pattern Recogn 43(1):5–13
13. Golub T, Slonim D, Tamayo P, Huard C, Gaasenbeek M, Mesirov J, Coller H, Loh M, Downing J, Caligiuri M, Bloomfield C, Lander ES (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286(5436):531–537
14. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182

15. Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Machine Learn 46(1-3):389–422
16. Hermes L, Buhmann JM (2000) Feature selection for support vector machines. In: 15th International Conference on Pattern Recognition, ICPR'00, Spain, pp 2712–2715
17. Huang C, Dun J (2008) A distributed PSO-SVM hybrid system with feature selection and parameter optimization. Appl Soft Comput 8(4):1381–1391
18. Jeong Y, Kang I, Jeong MK, Kong D (2012) A new feature selection method for one-class classification problems. IEEE Trans Syst Man, Cybern Part C 42(6):1500–1509
19. Khan J, Wei JS, Ringné M, Saal LH, Ladanyi M, Westermann F, Berthold F, Schwab M, Antonescu CR, Peterson C, Meltzer PS (2001) Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. Nat Med 7(6):673–679
20. Kittler J (1986) Feature selection and extraction. In: Handbook of Pattern Recognition and Image Processing. Orlando, FL: Academic Press, pp 59–83
21. Kohavi R, John GH (1997) Wrappers for feature subset selection. Artif Intell 97(1-2):273–324
22. Lashkia GV, Anthony L (2004) Relevant, irredundant feature selection and noisy example elimination. IEEE Trans Syst Man, Cybern Part B 34(2):888–897
23. Leardi R, Nørgaard L (2004) Sequential application of backward interval partial least squares and genetic algorithms for the selection of relevant spectral regions. J Chemometr 18(11):486–497
24. Lee D, Lee J (2007) Domain described support vector classifier for multi-classification problems. Pattern Recogn 40(1):41–51
25. Maldonado S, Weber R, Basak J (2011) Simultaneous feature selection and classification using kernel-penalized support vector machines. Inf Sci 181(1):115–128
26. Pomeroy S, Tamayo P, Gaasenbeek M, Sturla L, Angelo M, McLaughlin M, Kim J, Goumnerova L, Black P, Lau C, Allen J, Zagzag D, Olson J, Curran T, Wetmore C, Biegel J, Poggio T, Mukherjee S, Rifkin R, Califano A, Stolovitzky G, Louis D, Mesirov J, Lander E, Golub T (2002) Prediction of central nervous system embryonal tumour outcome based on gene expression. Nature 415(6870):436–442
27. Shao L, Liu L, Li X (2014) Feature learning for image classification via multiobjective genetic programming. IEEE Trans Neural Netw Learn Syst 25(7):1359–1371
28. Shieh M, Yang C (2008) Multiclass SVM-RFE for product form feature selection. Expert Syst Appl 35(1-2):531–541
29. Tax DMJ, Duin RPW (2004) Support vector data description. Mach Learn 54(1):45–66
30. Tayal A, Coleman TF, Li Y (2014) Primal explicit max margin feature selection for nonlinear support vector machines. Pattern Recogn 47(6):2153–2164
31. Wang J, Shan G, Zhang Q, DUAN X (2011) Research on feature selection method based on improved SVM-RFE. Microcomput Appl 32(2):70–74
32. Weston J, Mukherjee S, Chapelle O, Pontil M, Poggio TA, Vapnik V (2000) Feature selection for svms. In: Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, USA, pp 668–674
33. Xue B, Zhang M, Browne WN (2013) Particle swarm optimization for feature selection in classification: A multi-objective approach. IEEE T Cybern 43(6):1656–1671
34. Yang J, Ong CJ (2012) An effective feature selection method via mutual information estimation. IEEE Trans Syst Man, Cybern Part B 42(6):1550–1559
35. Yang W, Gao Y, Shi Y, Cao L (2015) MRM-Lasso: A sparse multiview feature selection method via low-rank analysis. IEEE Trans Neural Netw Learn Syst 26(11):2801–2815
36. Zhou X, Tuck DP (2007) MSVM-RFE: extensions of SVM-RFE for multiclass gene selection on DNA microarray data. Bioinformatics 23(9):1106–1114
37. Zhu Z, Ong Y, Dash M (2007) Markov blanket-embedded genetic algorithm for gene selection. Pattern Recogn 40(11):3236–3248

Li Zhang received the B.S. degree in 1997 and the Ph.D. degree in 2002 in electronic engineering from Xidian University, Xi'an, China. Now she is a full professor with the School of Computer Science and Technology, Soochow University, Suzhou, China. She was a postdoctor at the Institute of Automation, Shanghai Jiao Tong University, Shanghai, China, from 2003 to 2005. She worked as an associate professor at the Institute of Intelligent Information Processing, Xidian University, Xi'an, China, from 2005 to 2010. She was a visiting professor at Yuan Ze University, Taiwan, from February to May 2010. She has authored/co-authored more than 100 technical papers published in journals and conferences. Her research interests have been in the areas of machine learning, pattern recognition, neural networks and intelligent information processing.

Xingning Lu received the MS degree in 2017 at the School of Computer Science and Technology from Soochow University, Suzhou, China. She is now a support engineering in Microsoft, Wuxi, China. Her research interests have been in the areas of machine learning and pattern recognition.