Knowledge-Based Systems 42 (2013) 1–8

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

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

Rule extraction from support vector machines based on consistent region covering reduction

Pengfei Zhu [a,b], Qinghua Hu [a,*]

[a] School of Computer Science and Technology, Tianjin University, Tianjin 150001, China
[b] Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China

ARTICLE INFO

Article history:
Received 26 May 2012
Received in revised form 24 October 2012
Accepted 7 December 2012
Available online 8 January 2013

Keywords:
Classification learning
Rule extraction
Support vector machine
Consistent region
Covering reduction

ABSTRACT

Due to good performance in classification and regression, support vector machines have attracted much attention and become one of the most popular learning machines in last decade. As a black box, the support vector machine is difficult for users' understanding and explanation. In many application domains including medical diagnosis or credit scoring, understandability and interpretability are very important for the practicability of the learned models. To improve the comprehensibility of SVMs, we propose a rule extraction technique from support vector machines via analyzing the distribution of samples. We define the consistent region of samples in terms of classification boundary, and form a consistent region covering of the sample space. Then a covering reduction algorithm is developed for extracting compact representation of classes, thus a minimal set of decision rules is derived. Experiment analysis shows that the extracted models perform well in comparison with decision tree algorithms and other support vector machine rule extraction methods.

1. Introduction

Support vector machines (SVM) have attracted much attention from domains of machine learning and other applications due to their good performance [1–3]. They have been successfully applied in many fields, including bioinformatics [8], text classification [9], image recognition [10], fraud management [6], fault diagnosis [7], behavior prediction [47] and so on. Though SVM was initially developed for binary classification problems, researches on support vector machines have been devoted to deriving new techniques to solve classification problems with non-separable data [43], regression [44] and multi class problems [45,46]. Generally, SVM performs best among current classification techniques [1].

In spite of its excellent performance in classification and regression, especially in high-dimensional and continuously valued data, support vector machines, similar to artificial neural networks (ANNs), build black-box models that are not comprehensible to humans. Hence, the SVM and ANN do not provide an explanation or a comprehensible justification for the knowledge they learn, which has become one of the main obstacles for their further practical application [11–14]. Additionally, an explanation component related to the decision can be helpful for the acceptance of the technique by users. Sometimes this is badly expected in many domains such as medical diagnosis [12]. The classification techniques based on the construction of propositional if-then rules from relabeled data provide a completely transparent classification decision. However, it is hard to acquire the high classification performance and understandability simultaneously. If rules extracted from the SVM can provide classification performance equal to or better than the SVM and the comprehensibility is satisfied, the SVM model would be easily accepted by users.

To improve the comprehensibility of the SVM model, some efforts have been devoted to rule extraction from support vector machines. The techniques to learn rules from the SVM can be generally grouped into three categories: region based methods, learning based methods and methods based on support vectors.

In the region based techniques, regions with different shapes including ellipsoids [15], hyper rectangles [15–17] and hypercubes [12] are used to represent knowledge obtained from the SVM and the key issue is how to generate the region and define the boundary. In the related researches, Nunez et al. [15] proposed a rule extraction method based on support and prototype vectors to get regions including ellipsoids and hyperrectangles, in which prototype vectors are defined by a clustering technique. Similarly, Zhang et al. [16] generated hyperrectangles using prototype vectors for each class found by the support vector clustering algorithm and controlled the growth of the hyperrectangles by a nested generalized exemplar algorithm. In addition, Chen et al. [18] proposed a rule extraction algorithm by finding non-empty hypercubes for

* Corresponding author.
E-mail address: huqinghua@tju.edu.cn (Q.H. Hu).

0950-7051/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2012.12.003

gene expression data to improve comprehensibility of SVMs. For region based methods, in spite of high accuracy and fidelity, the number and the quality of rules are still affected by the initial parameters of the clustering algorithm [41].

As to learning based method, researches mainly create an artificial dataset based on the SVM model learned on the training dataset and then make use of the traditional rule learning methods such as C4.5 [19] and CART[20]to obtain rules from the artificial data. The main difference for the related work is how to generate the artificial dataset. In [21], the artificial data is generated by assigning the label decided by the SVM model to unlabeled samples. In [22], Martens et al. generates additional training examples close to randomly selected SVs and their labels are predicted by the SVM. The methods in this category extract relatively small number of rules with both high accuracy and fidelity. However, they do not open the black box and only use another classifier with better comprehensibility to model the outputs of the original SVM [41].

The methods based on support vectors rely on the information in the support vectors. Nahla et al. proposed SQRex-SVM to extract rules directly from a subset of the support vectors using a modified sequential covering algorithm [23]. Chaves et al. extract fuzzy rules from SVMs by projecting each feature in each of the SVs along its coordinate axes to form a number of fuzzy sets with triangular membership functions of the same length [24]. For the fuzzy rule extraction method, the extracted rules are of low accuracy and consistency [41].

In this paper, we propose a rule extraction technique based on consistent region covering reduct (named, CRCR_SVM) to figure out what the black box is and how it works. Firstly the consistent region of each sample in the training data set can be obtained according to the distance between the sample and the decision boundary. Then a suboptimal rule set is found from the leaned consistent region granules by consistent region covering reduction. In covering reduct, the volume maximization (VM) and point coverage maximization (PCM) criterions are used to select the most informative rule and the samples covered by this rule are deleted. After covering reduct, the pruning is done by choosing the first few rules that make the classification accuracy on the training or test data the highest. The quality of the rule set is evaluated by the classification accuracy, the number of rules and the fidelity. Experiments on both artificial data and real world data show that the proposed method can effectively improve the comprehensibility without foiling the classification performance and the result is comparable to or better than the traditional rule learning methods and SVM rule extraction techniques.

Different from the other region based methods, the proposed method is a non-parameterized model. Besides, the theoretic framework of consistent region covering reduction forms a mechanism for rule extraction. Thirdly,the proposed method can be applied to both linear SVM and nonlinear SVM.

The paper is organized as follows: Section 2 provides a brief introduction to the support vector machine. In Section 3, rule extraction from the SVM based on consistent region covering reduction is proposed. Experimental results on artificial and real world datasets are given in Section 4. In Section 5, the whole work is concluded.

2. Preliminaries

Support vector machines belong to large-margin classifiers, which try to find an optimal hyperplane that can separate the samples from different classes and maximize the classification margin.

Given a training set $S = \{x_i, y_i\},\ i = 1,\ \ldots,\ n,\ y_i \in \{-1, 1\},\ x_i \in \Re^m$. The SVM finds the optimal separating hyperplane with the largest margin [2]. Eqs. (1) and (2) represent the separating hyperplane in the linearly separable case:

$$wx_i + b \geqslant +1 \text{ for } y_i = +1 \tag{1}$$

$$wx_i + b \leqslant -1 \text{ for } y_i = -1 \tag{2}$$

For the linearly separable case, finding a maximum separating margin $d = 2/\|w\|$ is a constrained optimization problem represented by

$$\min \frac{1}{2}\|w\|^2 \text{ s.t. } \forall x_i \in S\ y_i(wx_i + b) \geqslant 1 \tag{3}$$

The hyperplane decision function can thus be written as:

$$f(x) = sign\left(\sum_{j=1}^{sv} a_j y_j (x_j, x) + b\right) \tag{4}$$

where (x_j, x) is the inner product of x_j and x.

In [34], Corinna et al. suggested a modified maximum margin idea (soft margin), which introduces slack variables ξ_i that measure the degree of misclassification [26]. With the regularization parameter C controlling the tradeoff between the margin and the training error [25], the optimization problem in Eq. (3) is then modified as follows:

$$\begin{aligned} \min \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & \forall x_i \in S, \quad y_i(wx_i + b) \geqslant 1 - \xi_i, \quad \xi_i \geqslant 0 \end{aligned} \tag{5}$$

where $\sum_{i=1}^{n}\xi_i$ is the upper bound on the training error and C is a regularization parameter [25].

In the case of nonlinear models, SVMs use kernel functions to map nonlinearly separable problems into a high-dimensional feature space, where tasks are linearly separable. In this way, a nonlinear classification problem can be linearly solved in the high-dimensional feature space [26,27]. The decision function of the kernel SVM is given by

$$f(x) = sign\left(\sum_{j=1}^{sv} \alpha_j y_j k(x_j, x) + b\right) \tag{6}$$

From Eq. (6), we can see that the SVM is a complex, non-linear function. It is difficult to understand the classification models in applications. However, the learned support vector machine model can provide us with useful information for extracting rules.

What information can we derive from a trained support vector machine? From Eq. (6), we know the decision function depends on support vectors, the kernel function, and coefficient α_j. In general, the support vectors are considered to be the most informative samples for the classification task. The classification hyperplane in feature spaces is the decision boundary, which can be used to identify the class region defined in the input or mapped space. Besides, the support vectors are the points closest to the classification hyperplane. Hence, they can also be used for establishing the borders of regions in the input space [12,15–17].

The trained SVM model can also be used to classify unlabeled artificial data and then rules can be generated by operating other direct rule learning methods on the artificial data [21,22]. Based on the components in the SVM model, the SVM can be comprehended by rule extraction.

3. Covering reduction based rule extraction from SVM

In this section, the whole rule learning process from the support vector machine is introduced. The main idea of the proposed method is given in Section 3.1. In Section 3.2, the initial rule set is learned by covering generation. Then a consistent region covering reduct technique is proposed to prune the initial rule set in Section 3.3. Finally, the classification principle based on the learned rule set is given in Section 3.4.

3.1. Basic idea

For rule extraction from SVM, firstly we should choose the type of the learning method and the form of rules. In our work, the proposed technique belongs to the region based methods and therefore the form of the rule is a region defined in the input space. The region should be consistent, that is, without differently labeled samples covered by this region. The size of the region is the distance from a sample to the hyperplane, as shown in the left part of Fig. 1. The shape of the regions in the previous work is restrained to one or two types [12,15–17]. In fact, the shapes can be different if diverse distance functions are utilized. By finding the consistent region for each sample, the initial rule set is formed.

Obviously the number of the regions is the same as the number of samples. Not all the regions are useful for classification. In fact, most regions could be redundant. Similar to other rule learning method, rule induction is needed to reduce useless rules and time complexity [4,5]. Which regions should be kept or removed? In Section 3.3, covering reduct is proposed to obtain a minimal rule set. After rule learning, given a query sample, its label could be obtained using the learned rule set. The learning process is shown in Fig. 2.

3.2. Consistent covering generation

In this section, we aim to find a proper and clean region for each sample. We define the consistent region of samples in terms of classification boundary, and form a consistent region covering of the sample space. Given a training data set $S = \{x_i, y_i\}$, $i = 1, \ldots, n$, $x_i \in \Re^m$. Considering a classification task, y_i is the class label of sample x_i. Formally, the data set can be written as $\langle U, A, D\rangle$, where $U = \{x_1, \ldots, x_n\}$, $A = \{a_1, \ldots, a_m\}$ is the set of attributes, $D = \{d\}$ is the decision attribute.

Definition 1. Given arbitrary $x_i \in U$, a subset of samples $\delta(x_i)$, called the neighborhood of x_i, is defined as [37]

$$\delta(x_i) = \{x_j \in U : \Delta(x_i, x_j) \leqslant \delta\}$$

where Δ is a distance function and δ is a parameter dependent on x_i.

Definition 2. Given $\langle U, A, D\rangle$, $x_i \in U$. $\delta(x_i)$ is said to be consistent and called consistent region if the samples in $\delta(x_i)$ belong to the same class. Furthermore it is called the maximal consistent region of x_i, denoted by $C(x_i)$, if $\delta(x_i)$ is consistent, and $\forall \delta' > \delta$, there is at least one sample in $\delta'(x_i)$ that comes from different classes.

Our aim is to extract rules that can be interpretable and used for classification. In region based methods, a region can be translated into a rule in that it contains the information of the center, the size and the label of the region. To keep the classification consistent, the region should be pure without samples of different classes, which is up to the size of δ. Moreover, we expect there are as few rules as possible in the decision model so that the model is simple and easy to understand. Hence, it is of great importance to choose a proper value for δ. So the definition of maximal consistent region is useful in extracting rules.

We set δ as the distance between samples and the classification hyperplane. If the hyperplane decision function is $f(x)$, then the distance d between sample x and the hyperplane is

$$d = \frac{|f(x)|}{\|w\|}$$

where $f(x)$ is the discriminative function. For linear SVM, $f(x) = \sum_{j=1}^{sv} \alpha_j y_j (x_j, x) + b$ and $w = \sum_{j=1}^{sv} \alpha_j y_j x_j$. For the nonlinear SVM, the discriminative function is $f(x) = \sum_{j=1}^{sv} \alpha_j y_j k(x_j, x) + b$, where $k(.,.)$ is a kernel function. In this case, $\|w\|$ can not be directly calculated. Computation of distance becomes a quadratic optimization problem with nonlinear constraint.

$$\text{minimized} d = d(x, \bar{x}) \text{ s.t. } \sum_{j=1}^{sv} \alpha_j y_j k(x_j, \bar{x}) + b = 0 \tag{7}$$

It is quite complex to solve this optimization. Besides, in most real-world applications, the classification tasks are inseparable. If the classification boundary is used, most regions could cover different labeled samples. Thirdly, as is shown in Fig. 1, the trained SVM classification plane is quite similar to the decision boundary formed by the regions whose size is the distance between the sample and its nearest similarly labeled support vector. In addition, the region based methods in [12,15–17] also approximated the decision boundary by the support vectors. Hence, we can use the support vectors to approximate the distance between the sample and the classification boundary. In this way, we can avoid solving the optimization problem in Eq. (7) and improve the consistency of the regions.

Note that as to each sample, a region can be formed. Hence, a set of regions, that is a set of consistent regions can form a covering of the input space. The family of consistent regions $C = \{C(x_1), C(x_2), \ldots, C(x_n)\}$generates a point-wise covering of the universe.

If the support vectors are $sv\_set = \{sv_1, \ldots, sv_l\}$, the size of consistent region of sample x_i is:

$$\delta = \Delta(x_i, sv_j)_{\min} \text{ where } j = 1, \ldots, l \text{ and } label(sv_j) = y_i \tag{8}$$

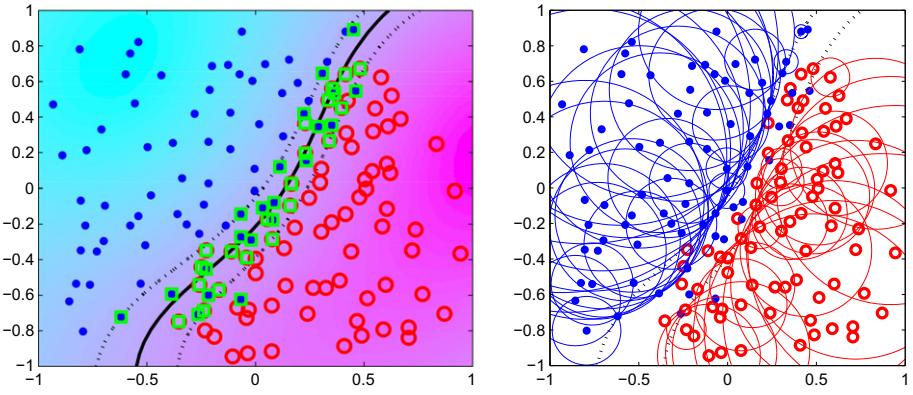

Fig. 1. The SVM classification plane (left) and the boundary approximated by the support vectors (right). dot and circle represent samples of two classes; square represents the support vectors; the solid line is the classification boundary. RBFSVM is used.

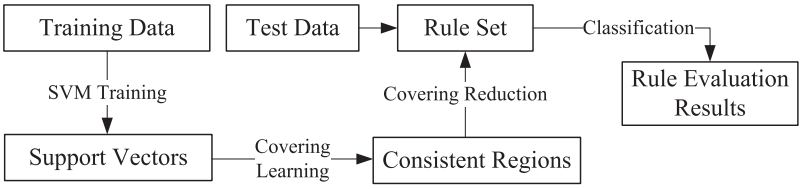

Fig. 2. Rule extraction from SVM based on consistent region covering reduction.

As for the distance function Δ, many functions are defined such as Euclidian Distance and Mahalanobis distance. In our work, a general distance function is used [39]:

$$d(x,y) = \left(\sum_{i=1}^{m} w_{a_i} \times d_{a_i}^{p}(x_{a_i}, y_{a_i})\right)^{\frac{1}{p}}$$

where m is the number of attributes, w_{a_i} is the weight of attribute a_i, and $d_{a_i}(x,y)$ is the distance between samples x and y with respect to attribute a_i.

It is obvious that p and w_{a_i} in the distance function can affect the shape of the consistent region covering. For example:

if $p = 2$ and $w_{a_i} = 1,\ i = 1,\ldots,m$, then the consistent region is a hypersphere;

if $p = 2$ and w_{a_i} varies for different a_i, then the consistent region is a hyperellipsoid;

if $p = \infty$ and $w_{a_i} = 1,\ i = 1,\ldots,m$, then the consistent region is a hypercube;

if $p = \infty$ and w_{a_i} varies for different a_i, then the consistent region is a hyperrectangle.

For kernel SVM, the distance between x and y is $\|\phi(x) - \phi(y)\|^2 = 2(1 - k(x,y))$, where $k(.,.)$ is a kernel function. Taking RBFSVM for example, the distance is $d(x,y) = 2 - 2 \times \exp(-\|x-y\|^2/2\sigma^2)$. According to the isotonicity of the distance, the location relationships of samples in Euclidean space and RBF kernel space are the same.

Each consistent region of a sample contains the information including the samples it covers and the size of the region. Here an information granule called consistent region granule is defined.

Definition 3. Given arbitrary $x_i \in U$, the consistent region granule $CRG(x_i)$ in feature space is defined as

$$CRG(x_i) = (x_i, y_i, \delta)$$

where δ is a parameter dependent on x_i.

A consist region granule corresponds to a rule. When we get n consistent region granules that cover all the samples, the initial rule set is found.

3.3. Consistent region covering reduction

After learning the consistent region covering, we get a set of consistent region granules. How can we find a minimal rule set from a set of consistent region granules? In this section, we try to find the rule set by consistent region covering reduction.

The family of consistent regions $C = \{C(x_1), C(x_2), \ldots, C(x_n)\}$ generates a point-wise covering of the universe. Now we call $\langle U,C,D\rangle$ a consistent region covering approximation space and $\langle U,C\rangle$ a consistent region decision system.

Definition 4. Let $\langle U,C,D\rangle$ be a consistent region covering decision system. X_i is one of the decision classes $C(x') \in C$. If $\exists C(x) \in C$, such that $C(x') \subseteq C(x) \subseteq X_i$, we say $C(x')$ is reducible consistent region with respect to X_i; otherwise, we say $C(x')$ is irreducible consistent region.

Definition 5. Let $\langle U,C,D\rangle$ be a consistent region covering decision system. If $\forall C(x) \in C$, there does not exist $C(x') \in C$, such that $C(x') \subseteq C(x) \subseteq X_i$, where $X_i$is an arbitrary decision class, then we say $\langle U,C,D\rangle$ is irreducible; otherwise, we say $\langle U,C,D\rangle$ is reducible.

Definition 6. Let $\langle U,C,D\rangle$ be a consistent region covering decision system. $C' \subseteq C$ is a derived covering from C by reducing the redundant covering regions, and $\langle U,C',D\rangle$ is irreducible. Then we say that C' is a D-relative reduct of C, denoted by $\text{reduct}_D(C)$.

Property 1. *Let $\langle U,C,D\rangle$ be a consistent region covering decision system and $\text{reduct}_D(C)$ is a D-relative reduct of C. Then $\langle U, \text{reduct}_D(C), D\rangle$ is also a consistent covering decision system, and $\forall C(x) \in C$, $\exists C(x') \in \text{reduct}_D(C)$, such that $C(x) \subseteq C(x')$.*

The conclusions of Property 1 are straightforward because we just remove the redundant covering regions in the covering. As all the elements in the consistent covering decision system are consistent, naturally the reduced covering decision system is also consistent.

We aim to find a a minimal reduction of the universe. After covering reduction, there is no redundant covering region. All the selected covering region granules are useful in approximating the decision classes.

The theoretic framework of consistent region covering reduction forms a mechanism for rule extraction. If we want to find a minimal rule set, we only need to find a minimal D-relative reduct from a set of consistent region granules. The minimal D-relative reduct is then transferred to a covering rule set.

Just like the problem of minimal attribute reduction, the search of minimal D-relative reduct set is also NP-hard [35,38]. There are several strategies to search the minimal D-relative reduct, such as forward search [40], backward search and genetic algorithm [36]. Here we consider the forward search technique which starts with an empty set of rules, and adds new rules one by one. Note that it is important to choose a proper evaluation criteria. In [5], J-measure is used to evaluate the rule and Jmax-pruning is proposed to improve Prism's classification accuracy. In [12] the volume maximization (VM) and point coverage maximization (PCM) criterions are used for rule evaluation. That is, the consistent region which has the largest size or covers most samples is selected and generates a piece of rule. In our work,VM and PCM criterions are adopted.

After covering reduction, the rule set still includes some rules that cover only a few samples. In this case, a further pruning strategy is needed to get a smaller rule set. Similar to the covering reduction, the consistent regions are ranked first by VM or PCM criterions. Then, the first few rules making the classification accuracy the highest on the training or test set are selected.

Given a training sample set $U\_Train = \{(x_1,y_1), \ldots, (x_n,y_n)\}$, $i = 1, 2, \ldots, n$ and a test sample set $U\_Test = \{(x_1,y_1), \ldots, (x_m,y_m)\}$, we can get a rule set $Rule = \{(x_1,y_1,\delta_1), \ldots, (x_h,y_h,\delta_h)\}$ by covering reduction. The algorithm is formulated as follows:

Algorithm 1. rule learning from the SVM based on consistent region covering reduct

1: **Procedure** INITIALIZE ($Rule \leftarrow \emptyset$)
2: compute the support vectors and the distance $\delta(i)$ between sample x_i, $i = 1, 2, \ldots, n$, and its nearest support vector
3: compute the consistent region $C(x_i)$ of sample x_i, $i = 1, 2, \ldots, n$, the covering of the universe is denoted by C. Remove $C(x)$, $card(C(x)) < 2$.
4: **While** $C \neq \emptyset$ **do**
5: select the consistent region $C(x)$ that covers the most samples or with the largest region size
6: add the rule (x, y, δ) generated by $C(x)$ to the rule set
7: remove the covering $C(x)$ from C
8: **end while**
9: Rank the rules in descending order
10: choose the first h rules that make the classification accuracy the highest on *U_Train* or *U_Test*
11:**end procedure**

This algorithm greedily searches the largest consistent region or the consist region that covers the most samples. In this way, we generate a small set. We also analyze the relationships between the size of consistent region and the number of covered samples in Section 4 to show why we select the most significant rule from two aspects.

For multi-class problem, as we get the support vectors of all the classes first by SVM training, the learned rule set can be directly applied to multi-class classification tasks. For SVM, the one against one (OAO) architecture is used.

3.4. Classification based on rule sets

When the rule set is obtained, it is a key issue to choose the classification principle. As to a query sample, there are three conditions for the number of rules that covers the sample. When the sample is covered only by one rule, the classification principle is quite definite. The label of the unlabeled sample is consistent with the rule that covers the sample. However, there are the other two cases that the sample is covered by no rule or more than one rule. In the case that a sample is not covered by any rule, Mitchell et al. propose that a default rule can be defined for classifying all the samples that are not covered by any rule [28]. Additionally, a similarity measure [42] can be used to assign the label of the nearest rule to the unlabeled sample [30]. With respect to the last case that a sample is covered by more than one rule, the classification can be done according to the frequency of the rule or the most specific rule [32]. Besides, the rules can also be ranked and the first rule covering the unknown sample would be chosen [29,31].

In our work, as to the case that a sample is covered by no rule or more than one rule, the label of the nearest rule is assigned to the query sample. The distance between a query sample and a rule is measured by the distance from the query sample to the center of the rule. For example, given a sample x and a rule $rule(i) = (x_i, y_i, \delta_i)$, the distance between x and $rule(i)$ is $d(x, x_i)$.

Given a new instance x and the rule set $Rule = \{(x_1, y_1, \delta_1), \ldots, (x_h, y_h, \delta_h)\}$, where N is the number of rules that covers the sample x, the classification rule is:

$$\text{if } N = 1, \Delta(x, rule(i)) < \delta_i, \text{ then } x \in rule(i) \text{ and } label(x) = y_i$$
$$\text{if } N = 0 \text{ or } N > 1, \ rule(i) \text{ is the nearest rule for } x, \text{ then } label(x) = y_i$$

4. Experimental analysis

The performance of the proposed rule extraction technique is evaluated on artificial and real word datasets. Firstly the proposed method is operated on an artificial dataset to show the whole rule extraction process. Secondly, the algorithm design is analyzed. We show the relationship between two rule evaluation criterions, i.e., VM (volume maximization) and PCM (point coverage maximization). In addtion, we also give the variance of the accuracy with selected rules added one by one to analyze the pruning technique. Then we compared the method with the original SVM in terms of classification accuracy. Finally, decision tree methods and other rule extraction techniques are compared to the proposed method. The code of CRCR_SVM can be downloaded from: http://www4.comp.polyu.edu.hk/~cspzhu/.

4.1. Artificial dataset

To intuitively show the rule extraction process from the SVM, an artificial data with 40 samples and two classes that satisfy gaussian distribution is generated. In Fig. 3 we can see that the left part is the distribution of the data in two-dimensional space. Then we obtain the support vectors by operating the SVM on the artificial dataset. By computing the distance between each sample and its nearest similarly labeled support vector, we can get the consistent region of each sample (the circles) in the middle part of Fig. 5. As is referred to in Section 3.2, a set of consistent regions forms a consistent covering of the input space. Based on covering reduction, a minimal rule set is got from the consistent region granules, as shown in the right of Fig. 3. Obviously we can only utilize three circles to represent the whole dataset. The experiment on the synthetic data shows that the proposed method can improve the

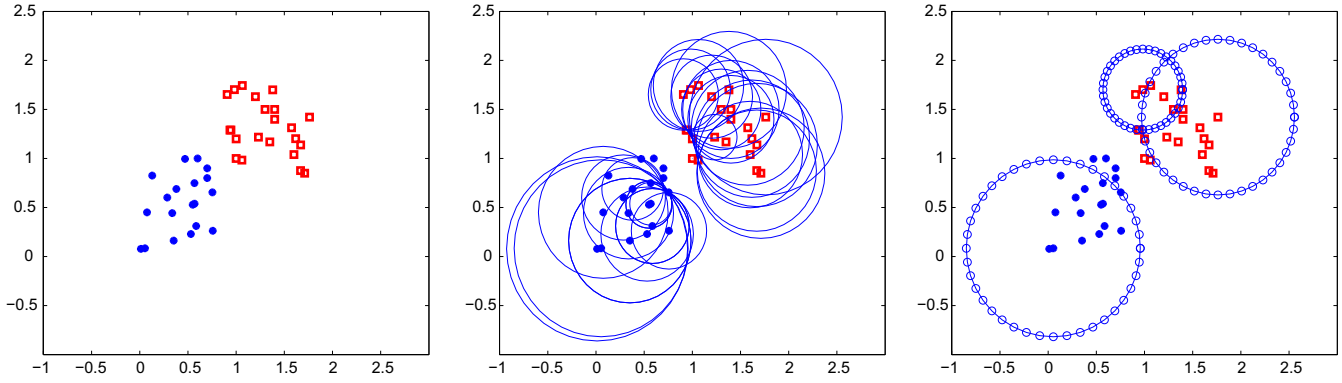

Fig. 3. Rule extraction process (circle).

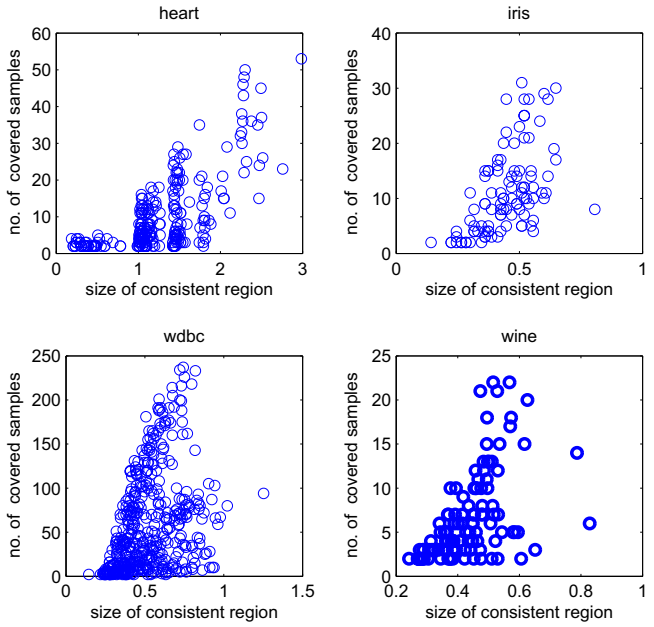

Fig. 4. Relationship between size of consistent region and the No. of samples in consistent region.

comprehensibility effectively by choosing consistent regions of some samples.

4.2. Algorithm analysis

In covering reduction, what type of consist region (rule) should be kept or removed? In our work, the consist region that covers the most samples (PCM) or with the largest size (VM) is chosen first and the samples in this consist region are removed. We calculate the relationships between the size of the consistent region and the number of samples in this consist region on four datasets. As is shown in Fig. 4, although the number of covered samples increases when the consistent region is larger, the relationships are not totally linear. Hence, we select the most significant rule from two aspects.

In rule extraction techniques, pruning is very important as the size of the rule set is still large and there are still some redundant rules covering only a few samples, which may result in overfitting. We operate CRCR_SVM on the training set and a rule set is obtained. Then the classification accuracy variance is shown in Fig. 5 as rules are added one by one. Note that test and training in Fig. 5 mean that rule pruning is done on the test set and training set respectively.

From Fig. 5 we can see that the classification accuracy increases at first and then goes down or does not vary when the rule set is too large. Hence, a pruning strategy is needed to select a smaller rule set. In our work, the first few rules that can make the classification accuracy the highest are selected.

4.3. Rule extraction result of CRCR_SVM and RBFSVM

We select ten datasets from university of California Irvine (UCI) repository [33] and the detailed description of the ten datasets is shown in Table 1. As linear SVM is relatively easy to understand, in this work, RBFSVM is chosen. Note that there is no constraint on the SVM types. The classification accuracy of RBFSVM is shown in the last column of Table 1. For σ and C (refer to Eq. (5)) in RBFSVM, they are automatically derived by cross validation.

The rule extraction quality is evaluated by three indexes including classification accuracy, number of rules and fidelity that

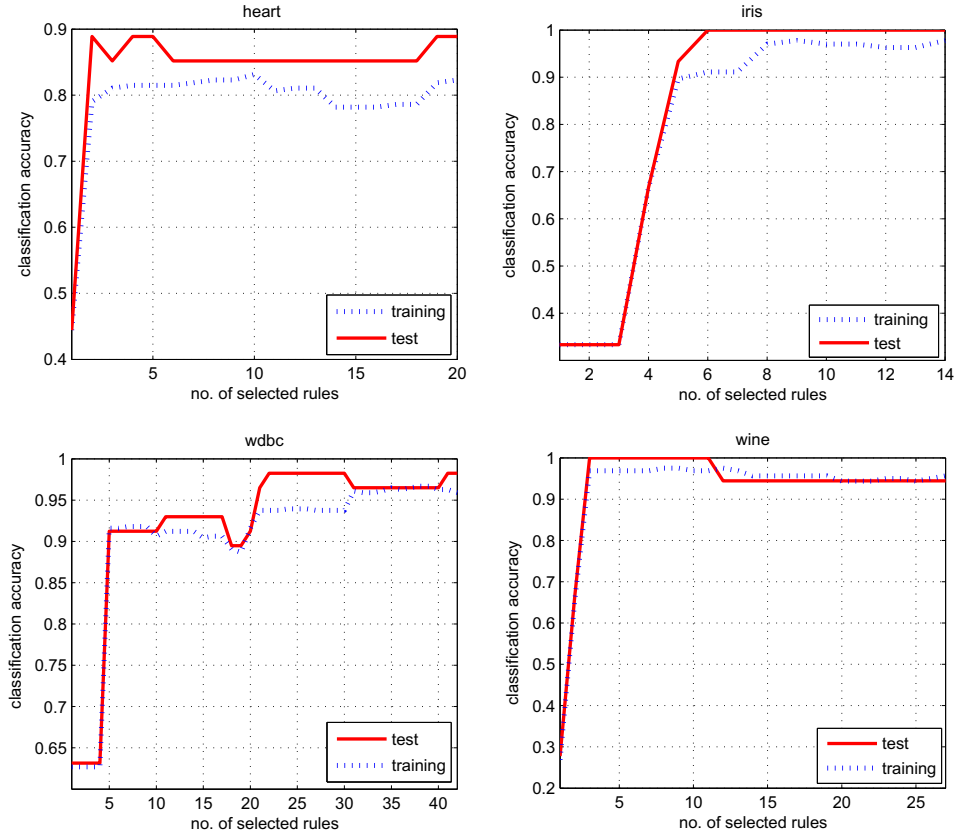

Fig. 5. Classification accuracy variance with the number of rules.

Table 1
Data description and classification accuracy of RBFSVM.

Data	Features	Class	Instances	RBFSVM
wine	14	3	178	98.9 ± 2.3
iris	5	3	150	96.7 ± 4.7
wdbc	31	2	569	95.8 ± 2.5
pima	9	2	768	76.7 ± 5.0
thyroid	6	3	215	95.8 ± 5.7
german	20	2	1000	70.1 ± 3.2
yeast	8	2	1484	76.5 ± 4.9
heart	14	2	270	83.0 ± 7.8
sick	30	2	2880	95.4 ± 0.5
wiscon	10	2	699	95.4 ± 3.9
average				88.4

reflects the extent of rule extraction mimicking the original SVM. The rule extraction result is shown in Tables 2 and 3. As a result of the difference in rule pruning, the experimental result consists of two parts. CRCR_SVM (PCM) and CRCR_SVM (VM) represent that, in covering reduct and the pruning strategy, rules are ranked by the number of covered samples (PCM) and the size of the region (VM) respectively. (Test) and (Training) represent that the pruning is done on the test dataset and training dataset respectively.

From Tables 1–3, we can see that the average classification accuracy of the proposed rule extraction technique is better than or on par with the original RBFSVM and we can use only a few rules to represent the whole dataset. In addition, the average fidelity is about 94%, which shows that the extent to which the proposed method mimics the original SVM model is very high. Besides, if we compare PCM and VM, the results in Tables 2 and 3 show that VM is better than PCM in terms of accuracy and the number of rules is comparable to each other on 9 out 10 datasets.

4.4. Comparison to direct rule learners

CRCR_SVM are further compared to the direct rule learners including C4.5, CART and Jripper, which are three of the most successful direct rule learners. The rule learning results are shown in Table 4, from which we can see that the proposed method are better than the three direct rule learners in terms of accuracy. When the pruning is done on the test set, the classification accuracy is about four percent higher than CART.

4.5. Comparison to other SVM rule extraction methods

We also compare the proposed rule extraction technique to other SVM rule extraction methods. In this part, the rule extraction technique based on prototypes and support vectors proposed by Nunez et al. [15] is compared. The experiment results in the literature[15] are used in this paper. From Table 5 we can see that CRCR_SVM is comparable to Nunez's method. When the pruning is done on the test set, the classification performance is better than Nunez's method. Besides, as the prototypes in Nunez's methods are obtained by clustering, the quality and number of rules are easily affected by the initial parameters [41]. As the proposed method is a nonparametric model, it is superior to other region based methods.

Table 2
Rule extraction result of CRCR_SVM (PCM).

Data	Circle (Test)			Circle (Training)		
	Accuracy	Num	Fidelity	Accuracy	Num	Fidelity
wine	98.9 ± 2.3	6.6	98.9	95.4 ± 5.7	18.8	95.4
iris	99.3 ± 2.1	10.1	96.7	96.7 ± 5.7	11.8	96.7
wdbc	97.6 ± 2.2	16.6	97.5	95.6 ± 2.4	27.9	96.7
pima	77.7 ± 3.2	40.8	90.4	75.8 ± 3.8	55.6	89.7
thyroid	92.6 ± 7.4	14.9	96.3	92.1 ± 7.7	18	95.8
german	74.0 ± 2.1	26.2	89.6	70.4 ± 2.0	30.3	86.1
yeast	77.7 ± 4.3	23.9	91.2	74.5 ± 3.8	81.2	91.6
heart	85.9 ± 4.9	6.5	94.4	81.9 ± 5.1	8.1	85.7
sick	94.5 ± 5.6	83.7	98.6	94.2 ± 5.1	118.5	98.0
wiscon	95.3 ± 3.2	18.3	97.0	94.4 ± 3.8	28.4	94.4
average	89.4	22.8	94.8	87.1	40.0	93.0

Table 3
Rule extraction result of CRCR_SVM (VM).

Data	Circle (Test)			Circle (Training)		
	Accuracy	Num	Fidelity	Accuracy	Num	Fidelity
wine	99.4 ± 1.8	9.1	98.9	95.4 ± 4.6	7.6	96.5
iris	98.0 ± 3.2	9.9	96.7	96.0 ± 5.6	14.6	96.0
wdbc	98.4 ± 1.7	13.5	97.4	95.6 ± 2.9	13.6	96.0
pima	83.1 ± 3.7	41.9	88.5	77.4 ± 4.4	45.4	89.7
thyroid	95.3 ± 7.6	9.7	98.2	94.4 ± 7.0	9.2	98.1
german	73.5 ± 2.0	71.2	88.3	71.1 ± 3.8	89	88.3
yeast	77.6 ± 3.8	43.9	93.1	75.0 ± 4.8	50.2	92.7
heart	85.7 ± 6.2	5.7	84.1	85.6 ± 4.8	23.2	85.2
sick	94.6 ± 4.0	145.7	98.7	94.6 ± 4.0	160.7	97.3
wiscon	97.7 ± 3.8	25.8	98.1	97.0 ± 3.8	25.7	98.0
average	90.3	37.6	94.2	88.2	43.9	93.8

Table 4
Rule learning results of CART C4.5 and Jripper.

Data	CART		C4.5		Jripper	
	Accuracy	Num	Accuracy	Num	Accuracy	Num
wine	88.2 ± 6.2	10	91.6 ± 7.3	5	93.8 ± 8.1	3
iris	95.3 ± 5.9	5	96.0 ± 5.0	5	94.0 ± 7.1	4
wdbc	93.1 ± 2.4	9	93.3 ± 3.8	13	94.3 ± 2.0	5
pima	74.4 ± 5.4	13	75.2 ± 7.4	20	75.0 ± 5.3	3
thyroid	90.1 ± 4.7	4	91.6 ± 3.6	9	92.5 ± 7.0	5
german	75.0 ± 3.2	10	71.7 ± 3.1	90	74.2 ± 2.5	3
yeast	76.1 ± 4.2	5	76.0 ± 3.9	44	75.0 ± 4.3	5
heart	78.5 ± 5.2	16	76.6 ± 4.8	18	78.8 ± 5.1	4
sick	98.5 ± 1.2	24	98.6 ± 1.3	15	98.2 ± 1.7	7
wiscon	93.1 ± 3.2	5	92.7 ± 3.6	55	94.4 ± 3.5	12
average	86.2	10.1	86.3	27.4	87.0	5.1

Table 5
Comparison with other SVM rule extraction methods.

Data	*Circle (Test)*			*Circle (Training)*		
	Accuracy	Num	Fidelity	Accuracy	Num	Fidelity
iris	98.0 ± 3.2	9.9	96.7	96.0 ± 5.6	14.6	96.0
wiscon	97.7 ± 3.8	25.8	98.1	97.0 ± 3.8	25.7	98.0
wine	99.4 ± 1.8	9.1	98.9	95.4 ± 4.6	7.6	96.5
thyroid	95.3 ± 7.6	9.7	98.2	94.4 ± 7.0	9.2	98.1
heart	85.7 ± 6.2	5.7	84.1	85.6 ± 4.8	23.2	85.2
average	95.2	12.0	95.2	93.7	16.0	94.8
	Nunez (Equation)			*Nunez (Interval)*		
iris	95.9	6.1	98.6	96.2	3.8	97.6
wiscon	96.1	15.3	98.6	95.9	14.5	96.5
wine	98.2	5.9	98.4	97.7	8.9	97.8
thyroid	95.1	7.3	97.1	95.3	10.8	95.3
heart	84.2	6.7	96.9	84.5	18.7	96.4
average	93.9	8.3	97.9	93.9	11.3	96.7

5. Conclusions

The support vector machine is one of the most successful learning machines and performs well especially in high dimensional and continuous data. However, the black-box model of SVM lack expla-

nation capacity, which is a main obstacle for the future application in some practical fields. In this paper, we proposed a rule extraction method from the SVM to figure out what the black box is and how it works. The consistent region is formed by computing the distance between the sample and the nearest support vector within the same class. Then a rule set is found based on the consistent region covering reduction. Experiment results on the artificial and real word datasets show that the proposed method can improve the comprehensibility effectively without foiling the classification performance compared to the original SVM and is comparable to the direct learning methods and other SVM rule extraction techniques.

Acknowledgment

This work is supported by Major State Basic Research Development Program (2013CB329304) and the National Natural Science Found for Excellent Young Scholars of China under Grant 61222210.

References

[1] V. Cherkassky, F. Mulier, Learning From Data, John Wiley & Sons, Inc., 1998.
[2] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge University Press, 2000.
[3] V. Vapnik, Statistical Learning Theory, John Wiley& Sons, Inc., 1998.
[4] F. Stahl, M. Bramer, Computationally efficient induction of classification rules with the PMCRI and J-PMCRI frameworks, Knowledge-Based Systems 35 (2012) 49–63.
[5] F. Stahl, M. Bramer, Jmax-pruning: a facility for the information theoretic pruning of modular classification rules, Knowledge-Based Systems 29 (2012) 12–29.
[6] P.F. Pai, M.F. Hsu, M.C. Wang, A support vector machine-based model for detecting top management fraud, Knowledge-Based Systems 24 (2) (2011) 314–321.
[7] X. Tang, L. Zhuang, J. Cai, C. Li, Multi-fault classification based on support vector machine trained by chaos particle swarm optimization, Knowledge-Based Systems 23 (5) (2010) 486–490.
[8] Chris Ding, Inna Dubchak, Multi-class protein fold recognition using support vector machines and neural networks, Bioinformatics 17 (2001) 349–358.
[9] S. Tong, D. Koller, Support vector machine active learning with applications to text classification, Journal of Machine Learning Research 2 (2002) 45–66.
[10] S. Li, H. Wu, D. Wan, J. Zhu, An effective feature selection method for hyperspectral image classification based on genetic algorithm and support vector machine, Knowledge-Based Systems 24 (1) (2011) 40–48.
[11] J. Ren, ANN vs. SVM: which one performs better in classification of MCCs in mammogram imaging, Knowledge-Based Systems 26 (2012) 144–153.
[12] G. Fung, S. Sandilya, R. Rao, Rule extraction from linear support vector machines, in: Proc. 11th Int'l Conf. Knowledge Discovery and Data Mining, 2005.
[13] G. Fung, S. Sandilya, R. Rao, Rule extraction from linear support vector machines, in: Proceedings of the Eleventh SIGKDD International Conference on Knowledge Discovery and Data Mining, 2005.
[14] R. Andrews, J. Diederich, A. Tickle, A survey and critique of techniques for extracting rules from trained artificial neural networks, Knowledge Based Systems 8 (1995) 373–389.
[15] H. Nunez, C. Angulo, A. Catala, Rule-extraction from support vector machines, in: Proc. European Symp. Artificial Neural Networks, 2002, pp. 107–112.
[16] Y. Zhang, H. Su, T. Jia, J. Chu, Rule extraction from trained support vector machines, in: Proc. Ninth Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining, 2005, pp. 61–70.
[17] X. Fu, C. Ongt, S. Keerthit, G. Hung, L. Goh, Extracting the knowledge embedded in support vector machines, in: Proc. IEEE Int'l Conf. Neural Networks, 2004, pp. 291–296.
[18] Z. Chen, J. Li, L. Wei, A multiple kernel support vector machine scheme for feature selection and rule extraction from gene expression data of cancer tissue, Artificial Intelligence in Medicine 41 (2007) 161–175.
[19] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, SanMateo, CA, 1993.
[20] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and Regression Trees, Wadsworth and Brooks, Monterrey, CA, 1984.
[21] N. Barakat, J. Diederich, Learning-based rule-extraction from support vector machines: performance on benchmark data sets, in: N. Kasabov, Z.S.H. Chan (Eds.), Proc. Conf. Neuro-Computing and Evolving Intelligence, 2004.
[22] D. Martens, B. Baesens, T.V. Gestel, Decompositional rule extraction from support vector machines by active learning, IEEE Transactions on Knowledge and Data Engineering 21 (2009) 177–190.
[23] N. Barakat, A.P. Bradley, Rule extraction from support vector machines: a sequential covering approach, IEEE Transactions on Knowledge and Data Engineering 19 (2007) 729–741.
[24] A.C. Chaves, M. Vellasco, R. Tanscheit, Fuzzy rule extraction from support vector machines, in: Proceedings of the Fifth International Conference on Hybrid Intelligent Systems, 2005.
[25] Q. Wu, D.X. Zhou, SVM soft margin classifiers: linear programming versus quadratic programming, Neural Computation 17 (5) (2005) 1160–1187.
[26] H. Byun, S.-W. Lee, Applications of support vector machines for pattern recognition: a survey, in: S.-W. Lee, A. Verri (Eds.), Lecture Notes in Computer Science, 2002, pp. 213–236.
[27] B. Schoelkopf, A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, Cambridge, 2002.
[28] T. Mitchell, Machine Learning, McGraw-Hill, 1997.
[29] I. Witten, E. y Frank, Data Mining, Practical Machine Learning Tools and Techniques with Java Implementations, second ed., Morgan Kaufmann Publishers, 2005.
[30] P. Domingos, Unifying instance-based and rule-based induction, Machine Learning 24 (1991) 141–168.
[31] M. Berthold, D. Hand, Intelligent Data Analysis an Introduction, Springer-Verlag, 1999.
[32] S. Salzberg, A nearest hyper rectangle learning method, Machine Learning 6 (1991) 251–276.
[33] C.L. Blake, C.J. Merz, UCI Repository of Machine Learning Data-Bases. University of California, Irvine. Dept. of Information and Computer Science, 1998. <http://www.ics.uci.edu/?mlearn/MLRepository.html>.
[34] Cortes Corinna, Vapnik Vladimir, Support-vector networks, Machine Learning 20(3) (1995) 273–297.
[35] T.L. Andersen, T.R. Martinez, NP-completeness of minimum rule sets, in: Proceedings of the 10th International Symposium on Computer and Information Sciences, 1995, pp. 411–418.
[36] M.J. Moshkov, A. Skowron, Z. Suraj, On minimal rule sets for almost all binary information systems, Fundamenta Informaticae 80 (2008) 247–258.
[37] Q.H. Hu, D.R. Yu, J.F. Liu, C.X. Wu, Neighborhood rough set based heterogeneous feature subset selection, Information Sciences 178 (2008) 3577–3594.
[38] Q.H. Hu, D.R. Yu, Z.X. Xie, Neighborhood classifiers, Expert Systems with Applications 34 (2008) 866–876.
[39] D. Randall Wilson, Tony R. Martinez, Improved heterogeneous distance functions, Journal of Artificial Intelligence Research 6 (1997) 1–34.
[40] Y. Du, Q.H. Hu, P.F. Zhu, P.J. Ma, Rule extraction for classification based on neighborhood covering reduction, Information Sciences 181 (2011) 5457–5467.
[41] N. Barakat, A.P. Bradley, Rule extraction from support vector machines: a review, Neurocomputing 74 (1–3) (2010) 178–190.
[42] G. Serpen, M. Sabhnani, Measuring similarity in feature space of knowledge entailed by two separate rule sets, Knowledge-Based Systems 19 (1) (2006) 67–76.
[43] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (1995) 273–297.
[44] H. Ducker, C. Burges, L. Kaufman, A. Smola, V. Vapnik, Support vector regresion machines, NIPS, vol. 9, MIT Press, 1997, pp. 155–162.
[45] C. Angulo, A. Catal, A: K-SVCR. A multi-class support vector machines, in: Proc.of ECML'2000, Lecture Notes in Computer Sciences, 2000, pp. 31–38.
[46] J. Weston, C. Watkins, Support vector machines for multi-class pattern recognition, in: Proc. of ESANN'99, 1999, pp. 219–224.
[47] Z.Y. Chen, Z.P. Fan, Distributed customer behavior prediction using multiplex data: a collaborative MK-SVM approach, Knowledge-Based Systems 35 (2012) 111–119.