Knowledge-Based Systems 116 (2017) 58–73

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

# An efficient instance selection algorithm to reconstruct training set for support vector machine

Chuan Liu[a,*], Wenyong Wang[a], Meng Wang[a], Fengmao Lv[b], Martin Konan[a]

[a] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[b] The Big Data Research Center, University of Electronic Science and Technology of China, Chengdu 611731, China

ARTICLE INFO

*Article history:*
Received 31 July 2016
Revised 18 October 2016
Accepted 31 October 2016
Available online 2 November 2016

*Keywords:*
Support vector machine
Instance selection
Machine learning

ABSTRACT

Support vector machine is a classification model which has been widely used in many nonlinear and high dimensional pattern recognition problems. However, it is inefficient or impracticable to implement support vector machine in dealing with large scale training set due to its computational difficulties as well as the model complexity. In this paper, we study the support vector recognition problem mainly in the context of the reduction methods to reconstruct training set for support vector machine. We focus on the fact of uneven distribution of instances in the vector space to propose an efficient self-adaption instance selection algorithm from the viewpoint of geometry-based method. Also, we conduct an experimental study involving eleven different sizes of datasets from UCI repository for measuring the performance of the proposed algorithm as well as six competitive instance selection algorithms in terms of accuracy, reduction capabilities, and runtime. The extensive experimental results show that the proposed algorithm outperforms most of competitive algorithms due to its high efficiency and efficacy.



## 1. Introduction

With the exponential growth of online information, finding ways of organizing data efficiently and effectively has become an important issue. Hence, in recent years, machine learning methods provide some solutions and have achieved excellent performance in a wide variety of fields [1] such as data mining, handwritten recognition, information retrieval, face detection, social network, diseases recognition and so on.

Support Vector Machine (SVM) [2] is an effective machine learning method with a solid theoretical foundation. It achieves a high prediction accuracy by learning the optimal hyperplane from training set, which greatly simplifies the classification and regression problems. Generally, SVM has many excellent features, such as high robustness and generalization ability with a small number of samples. That is, SVM determines the final optimal hyperplane by the minority support vectors with taking into consideration the model complexity and learning ability.

However, training SVM on large datasets is a very slow process and has become a bottleneck, since the quadratic programming (QP) problem that implies high training time complexity $O(n^3)$ and space complexity $O(n^2)$ needs to be solved [1,3,4]. Therefore, speeding up the training of SVM is a notable and significant issue. Hence, many methods have been developed to reduce the high computational complexity of SVM training on large scale datasets, and the survey in literature [5] can be referred to for detailed introduction. The well-known techniques [5] are sequential minimal optimization (SMO) [6], chunking [2], decomposition [7], and sampling [8].

The first type of above mentioned approaches speed up the training process by dividing the original QP problem into small pieces to reduce the size of the whole QP problem, such as the methods proposed by Dong [1] and J. Platt [9]. Although this type of methods solve the memory requirement issue in huge amounts of data, it still costs long processing time since the time consumption is closely related to the number of instances. The second kind of approaches make use of low-rank approximation, such as greedy approximation [10], matrix decomposition [7] and so on. The performance of these techniques has been extensively examined; however theoretically these techniques do not necessarily have high efficiency. Also, this kind of methods are relatively expensive in term of computation resource consumption. The third group of approaches consist in scaling down the training set by selecting support vector candidates, thereby using a small subset to train SVM [11]. In fact, SVM only relies on a fraction of

* Corresponding author. Fax. +86 028 61830563.
*E-mail address:* liuchuan@uestc.edu.cn (C. Liu).

http://dx.doi.org/10.1016/j.knosys.2016.10.031

samples, i.e. support vectors, thus we can efficiently remove the non-support vectors without affecting the classification accuracy. Hence, instance selection methods are proved to be ones of the most direct and effective ways for SVM to solve large scale classification problems and have been an attractive topic for many researchers.

Although many instance selection methods have been developed to successfully accelerate the training process of SVM on large scale datasets, most of them appear to have some drawbacks. For example, Reduced SVM (RSVM) [8] and Random Sampling Algorithm (RSA) [12] use a random algorithm which leads to uncertain results, while KMSVM [13,14] using clustering technology performs poorly when the classes are overlapped. In this paper, we propose a new efficient instance selection algorithm to reconstruct training set, which solves many serious difficulties, such as lack of memory and long processing time suffered by the existing instance selection algorithms in face of millions of records in their common applications. The proposed algorithm, named as Shell Extraction (SE), extracts the useless instances from training set, thereby preserving the maximum of support vectors. That is, the proposed algorithm does not directly select the support vectors which are difficult to be identified, but extracts the instances which are not likely to be support vectors. Meanwhile, we can adjust the strength to reconstruct different size of subsets. It should be pointed out that there are many remarkable properties in the proposed algorithm, which are summarized as follows.

1. It obtains the higher classification accuracy than most of the competitive algorithms.
2. It has a linear time complexity.
3. It can easily deal with multi-class problem.
4. The reduction intensity can be easily adjusted by its inputting parameter.

The rest of the paper is organized as follows. Section 2 introduces a brief overview of instance selection methods for scaling down training set. Section 3 describes three geometry-based methods in detail. Section 4 explains the theory of the proposed algorithm and analyzes its complexity. The conducted experiments and results are presented and discussed in Section 5. Finally, conclusion and further work are given in Section 6.

## 2. Instance selection: a brief review

The essence of support vector machine is to find the optimal classification hyperplane. The optimal hyperplane balances a term of forcing these partition between class $A$ and $B$ to maximize the margin of separation. Considering a set of linear separable samples $(\mathbf{x}_i, y_i), i = 1, 2, \cdots N$, where $\mathbf{x}_i \in R^D$ is the features, and $y_i \in \{+1, -1\}$ is the corresponding label. The classification hyperplane equation in *D-dimensional* space is $\omega \cdot \mathbf{x} + b = 0$. According to the requirements of optimal hyperplane, the problem is transformed into the QP problem as follows:

$$min\ \Phi(\omega) = \frac{1}{2}||\omega||^2 \tag{1}$$

$$s.t.\ \ y_i[(\omega \cdot \mathbf{x}_i) + b] \geq 1, i = 1, 2, 3, \cdots, N.$$

If the training data is not linear separable, the formula above must be modified to allow the classification violation samples as below:

$$min\ \Phi(\omega, \xi) = \frac{1}{2}||\omega||^2 + C \cdot \left(\sum_{i=1}^{N} \xi_i\right) \tag{2}$$

$$s.t.\ \ y_i[(\omega \cdot \mathbf{x}_i) + b] \geq 1 - \xi_i, i = 1, 2, 3, \cdots, N$$

$$\xi_i \geq 0, i = 1, 2, \cdots, N.$$

By introducing Lagrange multipliers, the dual formula of this problem can be rewritten as follows:

$$\max\ W(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i \cdot \mathbf{x}_j\right) \tag{3}$$

$$s.t.\ \ 0 \leq \alpha_i \leq C,\ \ i = 1, 2, \cdots, N$$

$$\sum_{i=1}^{N} y_i \alpha_i = 0.$$

Solving the problem above, we obtain the classifier as follows:

$$f(\mathbf{x}) = sign\left(\sum_{i=1}^{N} \alpha_i y_i (\mathbf{x} \cdot \mathbf{x}_i) + b\right), \tag{4}$$

where $\alpha_i$ is the solution of QP problem. In fact, the samples with $\alpha_i > 0$ define the optimal separating hyperplane, and the samples corresponding to the equality $\alpha_i = 0$ are non-support vectors which do not affect the results of training SVM. Unfortunately, in the training set, the number of support vectors is far less than the number of non-support vectors which occupy large storage space and consume a large amount of computing resources without any help for classification. Therefore, in order to improve the training speed of SVM and reduce unnecessary waste of resources, Instance Selection (IS) is a kind of feasible method, which has attracted the attention of many researchers.

Instance selection [15], also known as Prototype Selection (PS) [16] or reduction techniques [17], aims to select a subset of samples from the original training set, and it has the capacity to choose relevant samples and remove noisy and/or redundant, without generating new artificial data which is frequently yielded by Prototype Generation (PG) or abstraction methods [18]. A wide variety of IS methods based on different models for different applications have been proposed [17,19,20]. They can be broadly divided into three groups: condensation, edition, and hybrid methods. The main difference of them is dependent on the type of search carried out by the IS methods, whether they seek to retain border points, central points, or both of them. Condensation methods aim to retain border points which are closer to decision boundaries. The intuition behind these methods is that internal points do not affect classification as much as border points, since the hyperplanes between classes are mainly decided by border points. Thus, internal points can be removed with relatively little effect on classification. Edition methods, which are considered the opposite of condensation techniques, obtain smoother boundaries with border points removed which should be seen as noise. That is, such algorithms do not remove internal points that do not necessarily contribute to the decision boundaries. The effect of edition methods is to improve the generalization accuracy in testing data. Hybrid methods try to compute a smallest subset $S$, which allows the removal of internal and border points based on criteria followed by the two previous strategies, in order to maintain or even increase the generalization accuracy in testing data. It should be pointed out that the reduction capability of condensation strategies is comparatively higher than edition methods, since there are fewer border points than internal ones in most of datasets.

Instance selection has been broadly applied in classification [21], regression [22] and time series prediction [23]. Usually, different problems should be dealt with different IS strategies. For instance, condensation methods are probably suitable for reduction for training SVM, while edition strategies are more suitable

for the scene of using k-Nearest Neighbors (kNN) classifier. As we know, kNN suffers from several drawbacks such as high storage requirement, low efficiency in classification response, and low noise tolerance. Thereby, the objective of reduction for kNN rule is to compute a small consistent subset. The concept of consistency is formally defined as below [21,24]: given a non empty set $X$, a subset $S$ of $X$ (i.e. $S \subseteq X$) is consistent with respect to $X$ if the nearest neighbor rule using $S$ as training set can correctly classify all instances in $X$. Thus, according to the definition of consistency, if we want to reconstruct a subset $S$ to be used as the training set of kNN rule, edition and hybrid strategies maybe a good choice. However, the objective of this paper is to discuss the reduction methods for training SVM, thus we will mainly focus on condensation strategies.

Under the condensation taxonomy framework, there are still many issues, such as the order of search and the technologies of employ, can be involved in the further definition of the taxonomy. When searching for a subset $S$ of instances from original training set $X$, there are several directions in which the search can proceed [19]: incremental, decremental and batch. The incremental search process starts with an empty subset $S$, and iteratively adds each instance in $X$ to the subset $S$ if this instance satisfies the predefined criteria. The Condensed Nearest Neighbor (CNN) [24] proposed by Hart is considered to be the oldest formal proposal under this scheme. After CNN proposed, Ullmann's CNN [25] appeared to be more successful than Hart's CNN. Later, Tomek's CNN [26] was presented to consider only points close to boundary and remove the disadvantages of CNN. After that, a two-stage algorithm named as Mutual Neighborhood Value (MNV) [27] was introduced, which uses the concept of mutual neighborhood for selecting samples. Recently, modified CNN [28], generalized CNN [29], fast CNN [30] and Prototype Selection based on Clustering (PSC) [31] have been proposed one by one. One advantage of these incremental schemes is that they are suitable for dealing with data streams or online learning. However, the disadvantage is that they are prone to errors unless more information is available, since little information can be obtained in the beginning. Furthermore, these algorithms mostly depend on the order of presentation of samples.

The decremental search begins with the original training set $X$, and then searches for instances to remove from S. Also, the order of presentation is important for this kind of methods. The Minimal Consistent Set (MCS) [32] selects the samples in the order of significance of their contribution for enabling the consistency property. It should be pointed out that MCS leads to an unique solution irrespective of the initial order of presentation of instances. The Selective Nearest Neighbor (SNN) [33] algorithm is a representative of decremental methods, which produces a Selective Subset (SS) that can be seen as a condition stronger than that of consistency in order to find an alternate method for approximating nearest neighbor decision surfaces. Generally, a subset $S$ of $X$ is a selective subset, if it satisfies the following criteria [33] that (i) subset $S$ is consistent, (ii) the distance between any sample and its nearest selective neighbor is less than the distance from the sample to any sample of the other class, and (iii) subset $S$ should be the smallest one that satisfies the criteria (i) and (ii). Based on the concept of SS, the Modified Selective Subset (MSS) [34] primarily tries to find a better approximation to decision boundaries associated with the SS. Unlike the incremental strategies, decremental schemes need all the training instances to be available for examination each time. Thus, one disadvantage of decremental schemes is their higher computational cost than incremental ones.

Another way to search condensation subset is in batch mode. That is, each sample should be checked before selecting any of them. Then, all the samples that satisfy the consistent criteria are retained together. Many algorithms fall into this category. For example, Patterns by Ordered Projections (POP) [35] is to calculate which set of patterns could be covered by a "pure" region and then eliminate those inside that are not establishing the boundaries. The improved kNN [36] aims at "sparsifing" dense homogeneous clusters of patterns of any single class. This implementation involves iteratively eliminating patterns which exhibit high attractive capacities. The template reduction kNN [37] is based on defining the chain list which is a sequence of nearest neighbors from alternating class. Then, the authors set a cutoff for the retained patterns based on the fact that patterns further down the chain are close to the classification boundary. Moreover, Shin et al. [38] proposed a Neighborhood Property-based Pattern Selection (NPPS) algorithm, which is divided into two steps. First, the label Entropy of each point is calculated according to the kNN points; then they remove the label Entropy less than the threshold value of the point. If the point is closer to the separation boundary, the more heterogeneous points in its neighbor points, the greater the label Entropy. Therefore, this algorithm will retain more points at the boundary and delete points away from separation boundary. However, there are different view on the role of boundary points for edition methods. The NNSVM [39] considered that the intermixed points in other classes have no effect on the decision plane of SVM and, additionally, lead to overfitting. Thus, it searches the nearest neighbor of each point in the training set. If the point and its nearest neighbor belong to the same class, the point is marked as "1". If they belong to different classes, the point is marked as "-1". Then, all points marked as "-1" will be removed.

In fact, most of the algorithms described above are based on nearest neighbor techniques, which usually take a lot of time to calculate the nearest neighbor of each point. Thus, they suffer from involving a higher time complexity. Apart from above neighborhood-based IS methods, we have observed that the algorithms proposed are usually employing different techniques which can be sampling-based, clustering-based, decision tree-based, evolutionary-based and geometry-based methods.

In order to reduce the training set, Balcazar et al. [12] proposed a random sampling algorithm to produce a subset from the whole training set. Based on the idea of this approach, some other methods have been presented including Ferragut's SSVM [40], Lee's RSVM [8] and so on. Although their experiments show that these methods are faster than the original SVM, the performance of these randomly sampling algorithms is uncertain since some support vectors may not be included in the randomly selected subset. Thus, in [41], the authors executed a guided random selection of samples, which increases the probability of border points being sampled.

Besides the randomized sampling algorithms, methods that consider more about data characteristics are proposed to select the effective training subset. For example, Lyhyaoui et al. [42] put forward an instance selection technique via clustering to construct support vector subset. This approach performs clustering algorithm on training set, and finds the nearest cluster center from the opposite class to get instances near the decision boundary. Similarly, Chen et al. [43] given a Multi-Class Instance Selection (MCIS) algorithm based on clustering to select instances from multi-class. However, this method is based on an impractical assumption of no possible class overlap. Also, M.B.Almeida et al. [13] presented a procedure called KMSVM which is based on k-means clustering to accelerate the training of SVM. Specifically, they make use of k-means to create clusters of samples in the training set, and then cluster formed only by instances that belong to the same class label can be disregard and only cluster center are used. Conversely, cluster with more than one class label are preserved and added to the reconstructed subset. The key idea behind this method is to preserve instances near the separation boundaries and disregard instances far away from them. However, a problem accompanying the use of k-means algorithm is the choice of the num-

ber of desired output clusters. Furthermore, different initial points of k-means will lead to a completely different partition. Thereby, the result of clustering-based technique is unstable. More related studies of clustering-based methods can be referred to Koggalage's fast SVM [44], Tsai's outliers detection and removal methodology [45], and Li's minimum enclosing ball approach [46].

Different from clustering-based methods which use distance or density measure, literature [41] used a decision tree to form partitions that are treated as clusters. In this case, a purity function such as Gini index or Entropy gain will be used to create a tree, thus we do not need to predefine the number of clusters. Also, literature [47] proposed a method that uses a decision tree to decompose a given data space and train SVMs on the decomposed subsets. Doing so, the decision tree is used to replace the clustering, which simplifies the clustering procedures and avoids the high complexity computation of clustering. However, it performs poorly when dealing with non-convex training set. Furthermore, it becomes difficult to create a classifier for a large number of features [48].

With the different ideas of partition, J.R Cano et al. [49] introduced a strategy that uses evolutionary algorithm in instance selection. Also, S. Garcia et al. [50] adopted evolutionary-based algorithm named as EGIS-CHC in imbalanced classification. Moreover, Kawulok et al. [3] gave a novel idea called Dynamically Adaptive Genetic Algorithm (DAGA) which dynamically determines the desired training set size without any prior information.

In addition, the more efficient approaches which analyze the data geometry to determine an appropriate subset of instance selection are proposed. For example, Linear Fuzzy Support Vector Machine (LFSVM) [51] afforded a method based on the idea of class centroid. The algorithm can fast pick out some training samples which are impossible support vectors. Similarly, Pre-Selection sample based on Class Centroid (PSCC) [52] and Vector Projection Support Vector Machine (VPSVM) [53] are also geometry-based algorithms making use of the centroid of class for cutting down training set. However, in order to deal with the multi-class problem, these methods must transform it into a large number of binary classification problems, which no doubt decreases the efficiency of the algorithm.

In order to tackle large scale dataset, the stratification strategy (i.e. divide and conquer strategy) [19] are often employed to split the training set into disjoint subsets with equal class distribution. For example, Garf et al. [54] proposed the Cascade SVM, in which the training set is divided into a number of subsets, then these subsets are optimized by multiple SVMs. Also, Wang et al. [55] provided a two-stage approach by first cleaning data based on a bundle of weak SVM classifiers, and then appending two informative pattern extraction algorithms. As we all know, IS helps data mining algorithms to process large scale dataset which makes the application of classical algorithms difficult. However, some of IS methods also suffer from high time and space complexity, which make them not suitable for "Big Data" scenario. Hence, the use of stratification strategy also allows us to run any IS method over the disjoint subsets of the entire training set, thereby easing the problem of dealing with very large training set by reducing the number of instances that IS must handle simultaneously. However, stratification strategy can not reduce the high computational cost of these IS methods.

## 3. Geometry-based methods

We give further details of the representative geometry-based methods used in the experiments. These approaches analyze the data geometry to determine an appropriate training subset. Actually, the proposed method also belongs to this group of methods.

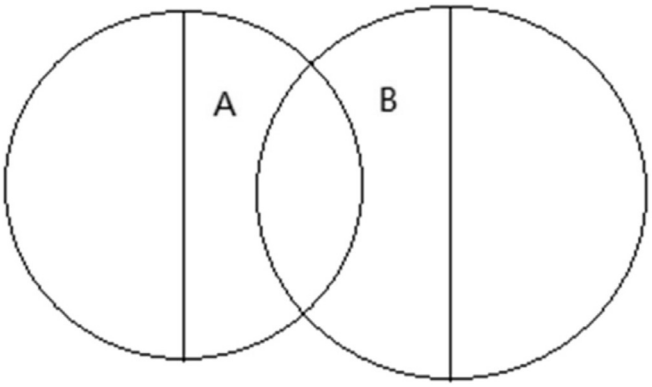



**Fig. 1.** Sketch of LFSVM method based on two adjacent spheres.

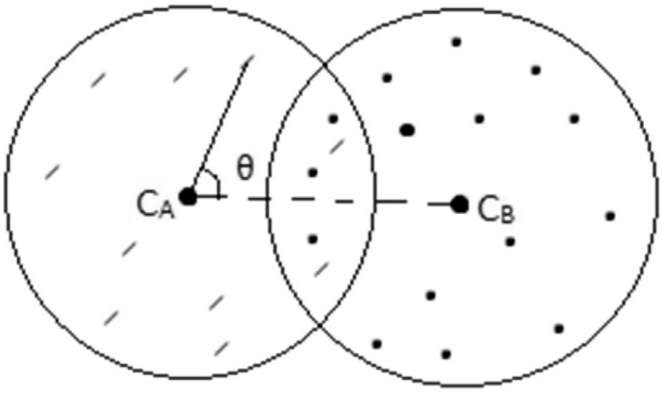



**Fig. 2.** Sketch of PSCC method based on two adjacent spheres.

The Linear Fuzzy Support Vector Machine (LFSVM) proposed by Cao et al. [51] is a typical geometric instance selection algorithm. The key idea is that the class distribution is assumed to be spherical, and the decision plane is distributed between the two spheres. Therefore, they suggest that the vectors distributed in the adjacent two hemispheres are the support vectors, as shown in Fig. 1. And other vector points, which are non-support vectors, can be deleted. The specific steps of the algorithm are as follows.

1. The centroid of the class is defined as follows:

$$\mathbf{c}_A = \frac{\sum_{i=1}^{N_A} \mathbf{x}_i}{N_A}, \tag{5}$$

   where $\mathbf{x}_i, i \in [1, N_A]$ are the points in class $A$, $N_A$ is the number of points in class $A$.
2. For each class, all points that satisfy the condition $(\mathbf{c}_A - \mathbf{c}_B) \cdot (\mathbf{x}_i - \mathbf{c}_B) \geq 0$ are kept, here $\mathbf{x}_i$ is the point in class $B$, $\mathbf{c}_B$ is the centroid of class $B$, and $\mathbf{c}_A$ the centroid of class $A$.
3. SVM is trained with the retained points.

On the basis of LFSVM, Luo et al. [52] proposed an adjustable reduction strategy named as Pre-Selection sample based on Class Centroid (PSCC). The main difference between this strategy and LFSVM is to introduce the cosine property of sample, as shown in Fig. 2. The cosine value of sample $\mathbf{x}_i$ in class $A$ is calculated as below:

$$\cos\theta = \frac{\overrightarrow{\mathbf{c}_A\mathbf{c}_B} \cdot \overrightarrow{\mathbf{c}_A\mathbf{x}_i}}{\left|\left|\overrightarrow{\mathbf{c}_A\mathbf{c}_B}\right|\right| \times \left|\left|\overrightarrow{\mathbf{c}_A\mathbf{x}_i}\right|\right|}. \tag{6}$$

Then, the sample will be removed from the original set if its cosine value is more than the threshold (denoted by $\varepsilon$) given by user. Thus, the reduction intensity can be adjusted according to the needs of user in PSCC. Therefore, PSCC can be seen as an upgraded

version of LFSVM. The concrete steps of this algorithm are as follows.

1. The centroid of each class is calculated with Formula (5).
2. The cosine value of each sample is calculated using Formula (6).
3. The sample whose cosine value is less than the given threshold is selected to train SVM.

Similar to cosine measure, the Vector Projection Support Vector Machine (VPSVM) method proposed in literature [53] selects sample on the basis of its projection measure. The projection of sample $\mathbf{x}_i$ is calculated as follows:

$$p_i=\frac{\overrightarrow{\mathbf{c}_A\,\mathbf{c}_B}\cdot\overrightarrow{\mathbf{c}_A\,\mathbf{x}_i}}{\left|\left|\overrightarrow{\mathbf{c}_A\,\mathbf{c}_B}\right|\right|}. \tag{7}$$

And the distribution radius of class $A$ is defined as below:

$$r_A=\max||\mathbf{x}_i-\mathbf{c}_A||. \tag{8}$$

Then, the details of the algorithm are presented as follows.

1. The centroid of each class is calculated by Formula (5).
2. The projection of each sample is obtained by Formula (7).
3. The distribution radius of each class is calculated by Formula (8).
4. The point will be retained to train SVM if the following conditions are met. If $r_A+r_B<\left|\left|\overrightarrow{\mathbf{c}_A\,\mathbf{c}_B}\right|\right|$, and the projection $p_i$ of $\mathbf{x}_i$ in class $A$ meets the condition $r_A-\delta\le p_i\le r_A$; if $r_A+r_B\ge\left|\left|\overrightarrow{\mathbf{c}_A\,\mathbf{c}_B}\right|\right|$, and the projection $p_i$ of $\mathbf{x}_i$ in class $A$ satisfies the condition $\left|\left|\overrightarrow{\mathbf{c}_A\,\mathbf{c}_B}\right|\right|-r_B-\delta\le p_i\le r_A+\delta$. Note that $\delta=\mu*\left|\left|\overrightarrow{\mathbf{c}_A\,\mathbf{c}_B}\right|\right|+\frac{F}{N}$ ($\mu$ indicates the ability of boundary area covering support vectors, $F$ is noise factor, and $N$ is number of samples), where $\mu$ and $F$ are given by user.

## 4. Shell extraction

In this section, the principle of the proposed algorithm is introduced. Subsequently, the detailed algorithm steps are presented and followed by the pseudo codes. Finally, the complexity of the presented algorithm is analyzed.

Given training set $X=\{\mathbf{x}_n: n=1,\cdots,N;\mathbf{x}_n\in R^D\}$, and the corresponding label set $Y=\{y_n: n=1,\cdots,N;y_n\in\{1,\cdots,M\}\}$, where $M$, $N$ and $D$ are respectively the number of classes, of samples and of features, and each sample has only one category label. Suppose a collection of samples with the same class label being $C_m=\{\mathbf{x}_n|y_n=m\}, m\in\{1,\cdots,M\}$, and $\mathbf{c}_m$ is centroid vector of $C_m$. The target of the proposed algorithm is to remove the non-support vectors (i.e. the samples corresponding to the equality $\alpha_n=0$ in SVM) from the training set.

As pointed out by Chen [43] that positive instances far away from centers of positive class and negative instances close to these centers are near the boundary. That is, the support vectors are mostly distributed in the boundary area, which are far away from the centroid point of each class. However, it is inadvisable to consider an instance to be support vector on the basis of the absolute distance between the instance and its centroid, since the instance close to its centroid is also likely to be support vector if the centroid is not located in the geometric center of this class due to the uneven distribution of instances in the vector space. Thus, the support vector can not be extracted directly according to the distance between this vector and its centroid. In contrast, a large number of non-support vectors, which are mostly close to the centroid of each class, are easily identified. Thus, the proposed algorithm is to identify non-support vectors based on the characteristics of the actual point distribution in the vector space. In the following, the proposed algorithm is firstly exposited in the vector space of non uniformly distributed instances, then it will be discussed in the uniformly distributed circumstance.

Supposing that the distribution of class $C_m$ is not uniform, which leads to the centroid not in the geometric center of this class, as shown in Fig. 3a. Also, we can mainly find out that the shape of support vectors in bold symbol is irregular. Thus, it is difficult to extract the support vectors directly according to the distance to the centroid. Alternatively, we can obtain the support vectors indirectly by deleting the non-support vectors, if the extraction of non-support vectors is easier and more efficient. Actually, we can easily obtain and delete the vectors in the round area, whose center is chosen as the centroid point of this class as shown in Fig. 3a. This round area is referred as Reduction Sphere (RS) in high-dimensional vector space. Obviously, a large number of non-support vectors will be retained if the radius (i.e. $R$) of RS is too small, while the support vectors near the centroid will be removed if the radius of RS is simply increased as illustrated in Fig. 3a.

Recall that the distribution of vectors in each class is not uniform as mentioned above, which means the new centroid point of this class will move to another location after the vectors in RS have been deleted. That is, the new centroid $\mathbf{c}_m'$ does not overlap with the old centroid $\mathbf{c}_m$ (kindly see below for proof), as shown in Fig. 3b. In fact, the new centroid will first move to the sparse area of vector distribution, and then come back to the dense area. Thus, as the moving of new centroid, the new RS can be iteratively created by using the new centroid as the center of this RS, and the vectors in it can be deleted. In this case, it need not initialize a large radius of RS in which some of the support vectors may be contained. Generally, different radii should be used in different categories. For category $C_m$, the radius can be calculated as below:

$$R_m=\frac{\lambda}{|C_m|}\sum_{\mathbf{x}_n\in C_m}dis(\mathbf{c}_m,\mathbf{x}_n), \tag{9}$$

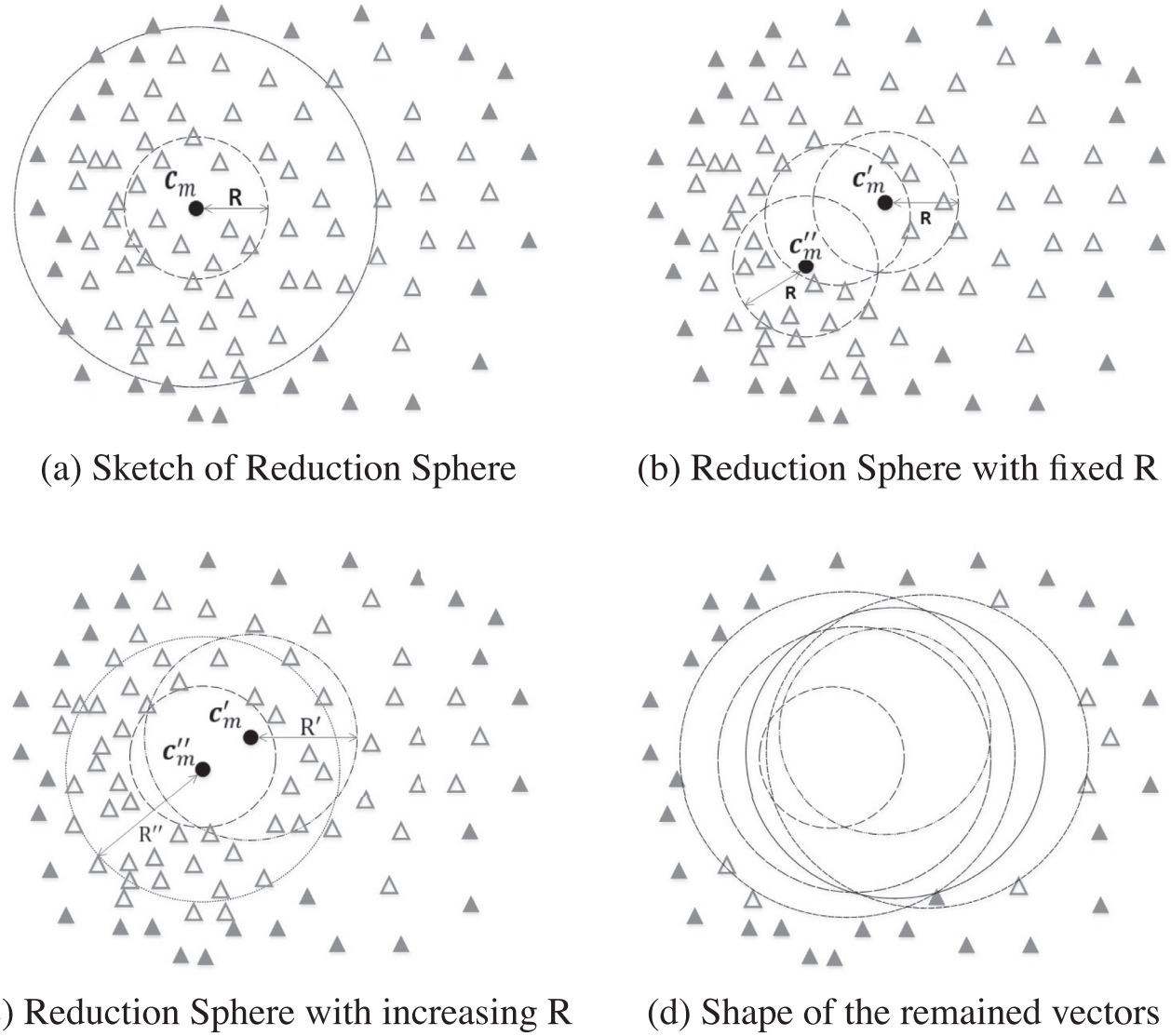
where $\lambda$ is the parameter given by user to control the radius, $|C_m|$ is the number of instances in category $C_m$, and $dis(\cdot,\cdot)$ is the measure of distance. However, it is accompanied with a problem that the algorithm usually stops before the required number of vectors has been removed if we use a fixed small radius. In order to deal with this problem, we iteratively increase the radius of RS in our algorithm as illustrated in Fig. 3c, until the number of retained vectors falls to the threshold denoted by $T_m=(1-\xi)*|C_m|$, where $\xi$ is the deleting percent given by user. Thus, the radius can be updated as follows:

$$R_m(i)=\frac{\lambda+\Phi(i)}{|C_m|}\sum_{\mathbf{x}_n\in C_m}dis(\mathbf{c}_m,\mathbf{x}_n), \tag{10}$$

where $\Phi(i)=\delta*i$ is a function of iteration number $i$, and $\delta$ is a factor given by user. Any other monotonically increase function is also acceptable. Finally, the mainly steps of Shell Extraction (SE) algorithm are summarized as follows:

1. Calculate the centroid $\mathbf{c}_m$.
2. Calculate the $dis(\mathbf{c}_m,\mathbf{x}_n)$ metric.
3. Delete the point $\mathbf{x}_n$ if $dis(\mathbf{c}_m,\mathbf{x}_n)<R_m(i)$, where $R_m(i)$ is computed by Formula (10).
4. Repeat step 2 to 3, until all points in this RS are deleted.
5. Repeat step 1 to 4, until the number of retained points falls to $T_m$.

With the centroid moving in each iteration, vectors in RS are constantly removed from the training set. Finally, the vectors distributed at the margin of each class are remained. Thus, the shape of the remained vectors is similar to a "Shell" in the high-dimensional space as illustrated in Fig. 3d. For multi-class training set $X$, the pseudo codes of SE algorithm is shown in Algorithm 1, where the parameter $\xi$ determines the strength of reduction. Thus, various size of training subsets can be produced by adjusting the




(a) Sketch of Reduction Sphere (b) Reduction Sphere with fixed R

(c) Reduction Sphere with increasing R (d) Shape of the remained vectors

**Fig. 3.** Schematic of Shell Extraction algorithm.

**Algorithm 1** Shell Extraction algorithm.

**Require:** $X$: the whole training set; $\lambda$: the parameter of initial RS radius; $\delta$: the parameter of RS increase; $\xi$: the deleting percent of training set.

**Ensure:** $X_S$: the reconstructed training set.

```
for m = 1; m ≤ M do
    T_m ← (1 − ξ) * |C_m|;
end for
Repeat i++;
for m = 1; m ≤ M do
    c_m ← 1/|C_m| Σ_{x_n ∈ C_m} x_n;
    R_m(i) ← (λ + δ*i)/|C_m| Σ_{x_n ∈ C_m} dis(c_m, x_n);
    for n = 1; n ≤ N && y_n = m do
        if |C_m| > T_m && dis(c_m, x_n) < R_m(i) then
            Delete x_n from C_m;
            Num++;
        end if
    end for
end for
Until N − Num ≤ Σ_{m=1}^{M} T_m;
Return X_S = X;
```

parameter $\xi$. Also, $\lambda$ and $\delta$ control the strength of the RS's movement. Thus, the parameters, i.e. $\lambda$ and $\delta$, have important impact on the performance of SE algorithm. Explicitly, the experiments conducted in Section 5 show that the optimal value of $\lambda$ falls into the region of [0.8,1] and $\delta$ should be set to be a small value. Generally, a large value of $\lambda$ (or $\delta$) may lead to some of support vectors being contained in RS, while a small value means a slight movement of RS and a large number of iterations.

SE algorithm in each iteration can be divided into two steps: finding the centroid whose time complexity is $O(ND)$ and calculating the distance between the points and the centroid that also has a time complexity equals to $O(ND)$. Therefore, Shell Extraction algorithm has a linear time complexity. As it is mentioned above, SE is based on the inference that new centroid will not overlap with the old one if the distribution of instances is uneven. Now we take the $k$th dimension of instance as an example, and the proof is as follows:

1. Suppose $c_{mk} = \frac{\sum_{n=1}^{N_m} x_{nk}}{N_m} \in [0,1]$, where $c_{mk}$ is the $k$th dimension of the centroid $\mathbf{c}_m$ and $N_m = |C_m|$.

2. Suppose $c'_{mk}=\frac{\sum_{n=1}^{N_m-N_d} x_{nk}}{N_m-N_d}\in[0,1]$, where $c'_{mk}$ is the $k$th dimension of the new centroid $\mathbf{c}'_m$ and $N_d$ is number of the deleted non-support vectors.
3. Assume $c_{mk}=c'_{mk}$, then $\frac{\sum_{n=1}^{N_m} x_{nk}}{N_m}=\frac{\sum_{n=1}^{N_m-N_d} x_{nk}}{N_m-N_d}\Longrightarrow\frac{\sum_{n=1}^{N_d} x_{nk}}{N_d}=\frac{\sum_{n=1}^{N_m} x_{nk}}{N_m}=c_{mk}$ $\Longrightarrow\sum_{n=1}^{N_d} x_{nk}=N_d*c_{mk}$.

Actually, the probability, i.e. $P\left(\sum_{n=1}^{N_d} x_{nk}=N_d*c_{mk}\right)$, of this conclusion is close to zero, when the distribution of instances is uneven. Therefore, this is in contradiction with the hypothesis. Furthermore, considering all features of instance, the probability of $\mathbf{c}'_m=\mathbf{c}_m$ is $\Pi_{k=1}^{D}\left[P\left(\sum_{n=1}^{N_d} x_{nk}=N_d*c_{mk}\right)\right]\approx 0$, which will not happen.

Recall that we made a hypothesis that the instances are unevenly distributed in the vector space. Conversely, if the distribution of instances is uniform, the origin centroid will be located in the geometric center of each class. In this case, the new centroid of each class will be fixed in the center position after the vectors in RS have been deleted. As the radius of RS grows iteratively, the non support vectors close to the center will continue to be removed. Therefore, SE algorithm can also work well in the vector space of uniform distribution.

## 5. Experiments

In order to justify the rationality of SE, we compare it with six representative IS methods, i.e. NNSVM, CNN, KMSVM, LFSVM, PSCC and VPSVM, based on a number of standard benchmark collections which come from UCI repository. Moreover, the parameters of SE are also analyzed in a two-dimensional artificial dataset. In our experiments, the items to be investigated are (i) the ability of keeping classification accuracy; (ii) the reduction ratio (the ratio is defined as the number of removed instances divided by the total number of instances) of each method; and (iii) the effect of parameters on SE. The experiments are performed on a PC with Intel(R) Pentium(R) CPU G2030 at 3.0 GHz 8 GB RAM, Windows 7, 64 bit Operating System. Note that the datasets[1] and source codes[2] associated with this paper are available on our Website.

### 5.1. Datasets

The experiments are carried out on eleven datasets including nine low dimensional and two high dimensional datasets. The low dimensional datasets are concisely introduced as follows. *Glass* comes from USA Forensic Science Service which has six types of glasses and is defined by terms of their oxide content. *Heart* disease dataset contains 4 classes, i.e. Cleveland, Hungary, Switzerland and VA Long Beach. The original database contains 76 attributes, but all published experiments refer to using a subset of 13 attributes. *Ionosphere* is the classification of radar returned from ionosphere. This radar data was collected by a system in Goose Bay, Labrador. The instances are described by 2 attributes per pulse number, corresponding to the complex values resulting from electromagnetic signal. *Dermatology* is used to determine the type of Erythematous-Squamous disease, which contains six classes and 34 attributes. *Segment* is an image segmentation database, of which the instances were drawn randomly from a database of seven outdoor images. The images were segmented to create a classification for every pixel. *Waveform* is CART book's waveform domain, which contains three classes and 21 attributes. *Isolet* is used to predict the letter-name spoken. *USPS* appears in the book "the elements of statistical learning" by Friedman [56]. *Letter* is a database of character image features, which is used to identify the letter.

**Table 1**
Summary of the information and the parameters for SVM in each dataset .

| Datasets | Siz. | Fea. | Cls. | s | t | c | g | n |
|---|---|---|---|---|---|---|---|---|
| *Dermatology* | 358 | 34 | 6 | 1 | 0 | $2^7$ | $10^{-4}$ | 0.1 |
| *Glass* | 214 | 9 | 6 | 0 | 1 | $2^7$ | $10^0$ | 0.2 |
| *Heart* | 270 | 13 | 2 | 0 | 1 | $2^{-2}$ | $10^{-1}$ | 0.1 |
| *Ionosphere* | 351 | 34 | 2 | 0 | 0 | $2^5$ | $10^{-2}$ | 0.2 |
| *Isolet* | 7797 | 617 | 26 | 0 | 1 | $2^{-2}$ | $10^{-2}$ | 0.1 |
| *Letter* | 20000 | 16 | 26 | 0 | 1 | $2^1$ | $10^{-2}$ | 0.1 |
| *Segment* | 2310 | 19 | 7 | 0 | 1 | $2^{-1}$ | $10^{-2}$ | 0.1 |
| *USPS* | 9298 | 256 | 10 | 0 | 2 | $2^7$ | $10^{-2}$ | 0.1 |
| *Waveform* | 5000 | 21 | 2 | 0 | 1 | $2^{-2}$ | $10^{-2}$ | 0.1 |
| *Newsgroup* | 13128 | 29949 | 20 | 0 | 2 | $2^7$ | $10^{-1}$ | 0.1 |
| *Reuters* | 9462 | 8455 | 58 | 0 | 2 | $2^7$ | $10^{-1}$ | 0.1 |

The high dimensional text collections are *20Newsgroups* and *Reuters-21578*. *20Newsgroups* contains 19,997 messages from 20 kinds of news, including 4% reprint. *Reuters-21578* is a group of 1987 reuters news. We combine the training and testing sets of the version of Apte's split 90 categories which contains 11,406 texts. For text collections, we delete the samples that have multiple labels and then remove the categories whose samples are less than ten, since we focus on single-label classification task. Then, a stop word list is used to remove common words, and the Porters stemming algorithm is adopted to compute the root of each word. Finally, Term Frequency-Inverse Document Frequency (TF-IDF) [57] weighting technology is used to transform the text into high dimensional vectors. The details, including size (Siz.), number of features (Fea.) and number of classes (Cls.), of each dataset are listed in Table 1 after the above pre-processing steps.

### 5.2. Setting of experiments

The investigated algorithms can be mainly divided into two categories. One group of algorithms can obtain different reduction ratio by adjusting their parameters, such as PSCC (adjusting $\varepsilon$), VPSVM (adjusting $\mu$ and $F$) and SE (adjusting $\xi$). And the other group of algorithms can not adjust the ratio, such as LFSVM, KMSVM, NNSVM and CNN. Therefore, we conduct the following two groups of experiments by first comparing SE with VPSVM and PSCC in different reduction ratios; and then we do the comparison with LFSVM, KMSVM, NNSVM and CNN in approximate reduction ratios. Doing so, it is worth noting that the so-called approximate reduction ratios refer to the difference between two ratios being in the range of 2% in general, since the ratio is controlled by adjusting the parameters and we can not guarantee to obtain the same ratio in the experiments. For SE, $\lambda$ is set to 0.8 in low dimensional datasets, and 1.0 in high dimensional datasets. Also, $\delta$ is chosen as 0.01. For KMSVM, the number of clusters is set to 20 in *Dermatology*, *Glass* and *Heart*, while the number is set to 200 in the remaining datasets.

For low dimensional datasets, we use SVM as the classification model; with respect to high dimensional datasets, we employ SVM and CBC (Centroid Based Classifier) [57]. *LIBSVM*[3] is used as the tool of SVM in all experiments. In order to achieve the best performance, we respectively debug the parameters, i.e. type (-s), type of kernel (-t), loss function (-c), gamma function (-g) and v-svc parameter (-n), of *LIBSVM*, where -s traverses 0, 1; -t traverses 0, 1, 2; -c traverses $2^{-7}, 2^{-6}, \ldots, 2^7$; -g traverses $10^{-1}, 10^{-2}, \ldots, 10^{-5}$; -n traverses $0.1, 0.2, \ldots, 0.5$. A total of 2250 experiments are carried out to obtain the best parameters of classification performance for each dataset, as also shown in Table 1.

[1] ftp://nepsnet.com:25601/.
[2] https://github.com/liuchuan-uestc/ISmethod.
[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

We choose $MacroF_1$ ($m\_F_1$) and $MicroF_1$ ($\mu\_F_1$) as the evaluation indices of classification. Supposing the number of samples classified correctly into class $C_m$ is $a$, the number of samples classified incorrectly into $C_m$ is $b$, and the number of samples in $C_m$ classified into other classes is $c$. The precision ($p_m$) and recall ($r_m$), which are used to evaluate the classification performance of class $C_m$, are defined as $a/(a+b)$ and $a/(a+c)$, respectively. Usually, there is an inverse relationship between precision and recall. Hence, $m\_F_1$ and $\mu\_F_1$ are used to measure the average performance for the whole categories, where $F_1$ is a weighted combination of precision and recall, and it is defined as:

$$F_1(r, p) = \frac{2pr}{p+r}.$$

Thereby, $m\_F_1$ is computed by averaging $F_1$ of all categories, while $\mu\_F_1$ is calculated by averaging the precision and recall of all instances. Since $m\_F_1$ gives the weight to all categories equally, it will mainly be influenced by the performance of rare categories. In contrast, $\mu\_F_1$ treats all instances equally, it will be dominated by the performance of common categories. Moreover, with respect to distance metric, the cosine distance is employed in the real datasets, while Euclidean distance is used in the artificial dataset.

There are two main methods for evaluating the performance of IS algorithms, i.e. hold-out and k-fold cross-validation methods. The hold-out method is unreliable since the selection of testing set has a direct impact on the whole performance. Hence, the five-fold cross-validation method is employed to avoid the influence of randomly selected testing set. That is, a given dataset is randomly splitted into five subsets. Each time one testing subset is selected to estimate the predictive performance of SVM, and the remaining subsets are used to train SVM after the IS algorithm has been applied on them. Then, averaging the results of multiple splitting is commonly used to decrease the variance of the estimation.

### 5.3. Experimental results and analyses

In the first group of experiments, SE is compared with VPSVM and PSCC in different ratios based on the low dimensional datasets as shown in Figs. 4 and 5, where Fig. 4 is the comparison in $\mu\_F_1$, and Fig. 5 is in $m\_F_1$. The abscissa is the ratio, and the ordinate is $\mu\_F_1$ or $m\_F_1$. In the following, we focus on the analysis of $\mu\_F_1$, since the trends of $\mu\_F_1$ and $m\_F_1$ are basically consistent.

It can be easily seen that SE has an outstanding performance in *Heart*(Fig. 4b), *Ionosphere*(Fig. 4c), *Dermatology*(Fig. 4d), *Segment*(Fig. 4e), *Isolet*(Fig. 4g), *USPS*(Fig. 4h) and *Letter*(Fig. 4i) datasets. For example, though the performance of SE is lower than PSCC when the ratio is 10% in *Heart*(Fig. 4b), SE quickly exceeds the other two algorithms when the ratio is greater than 20%. Moreover, SE maintains an upward trend and finally reaches the maximum value of 0.80 in the ratio of 70%. However, the curves of the other two algorithms keep declining and finally reach their lowest values in the ratio of 80%. At this time, $\mu\_F_1$ of SE is about 30 percent higher than the other two algorithms.

The performance of SE is lower than the other two algorithms in *Ionosphere*(Fig. 4c) when the ratio is less than 30%. However, the other two algorithms have the most serious deterioration when the ratio is more than 30%, while SE only has a small attenuation at this time. Explicitly, $\mu\_F_1$ of SE maintains at 0.82 in the ratio of 80%, while the other two algorithms are less than 0.40 at this time. That is, $\mu\_F_1$ of SE is at least 0.4 higher than the other two algorithms in the ratio of 80%.

Similarly, SE always has the superiority performance in *Dermatology*(Fig. 4d), *Segment* (Fig. 4e), *Isolet*(Fig. 4g), *USPS*(Fig. 4h) and *Letter* (Fig. 4i) datasets. For instance, in *Letter*(Fig. 4i), $\mu\_F_1$ of SE, PSCC and VPSVM are about 0.95 in the ratio of 10%; but $\mu\_F_1$ of SE still maintains at 0.79 in the ratio of 80%, while PSCC and VPSVM are about 0.5 and 0.42, respectively. In addition, it should be pointed out that SE shows a poor performance on two datasets, i.e. *Glass* (Fig. 4a) and *Waveform*(Fig. 4f).

**Table 2**
The comparison results of LFSVM and SE in each dataset.

| Datasets | Ratio | LFSVM $m\_F_1$ | LFSVM $\mu\_F_1$ | SE $m\_F_1$ | SE $\mu\_F_1$ |
|---|---|---|---|---|---|
| *Dermatology* | 49.79% | 0.875 | 0.893 | **0.941** | **0.948** |
| *Glass* | 59.58% | **0.541** | **0.561** | 0.403 | 0.508 |
| *Heart* | 50.37% | 0.544 | 0.541 | **0.758** | **0.742** |
| *Ionosphere* | 58.39% | 0.607 | 0.624 | **0.831** | **0.852** |
| *Isolet* | 54.30% | 0.950 | 0.949 | **0.968** | **0.958** |
| *Letter* | 52.95% | 0.783 | 0.777 | **0.903** | **0.894** |
| *Segment* | 54.95% | 0.783 | 0.780 | **0.920** | **0.922** |
| *USPS* | 55.47% | 0.974 | 0.977 | **0.978** | **0.980** |
| *Waveform* | 49.72% | **0.799** | 0.770 | 0.793 | **0.784** |
| *Newsgroup*-SVM | 40.74% | 0.863 | 0.872 | **0.934** | **0.933** |
| *Newsgroup*-CBC | 40.74% | 0.851 | 0.849 | **0.916** | **0.916** |
| *Reuters*-SVM | 38.23% | **0.455** | 0.723 | 0.438 | **0.725** |
| *Reuters*-CBC | 38.23% | 0.489 | 0.653 | **0.514** | **0.671** |

As shown in Fig. 6, SE is compared with PSCC and VPSVM in two high dimensional datasets. The results using SVM in *NewsGroup* are shown in Fig. 6a and b. In the beginning, SE has the similar performance with the other two algorithms. However, it is easy to find out the advantages of SE as the ratio's increasing. For instance, $\mu\_F_1$ of SE maintains at 0.93 in the ratio of 20%, which is about 5 percent higher than the other two algorithms. When the ratio reaches 80%, $\mu\_F_1$ of SE still maintains at 0.90, while $\mu\_F_1$ of PSCC is 0.51 and $\mu\_F_1$ of VPSVM is 0.45. Therefore, PSCC and VPSVM appear to significantly decline when the ratio is more than 50%, which does not happen on SE.

The results with CBC in *NewsGroup* are shown in Fig. 6c and d. The performance of SE remains unchanged with the rise of ratio. However, PSCC and VPSVM are declining from the beginning. Thus, the performance gap between SE and the other two algorithms is gradually increasing. At last, $\mu\_F_1$ of SE is about 25 and 40 percent higher than PSCC and VPSVM in the ratio of 80%, respectively.

The results using SVM in *Reuters* are shown in Fig. 6e and f, where $\mu\_F_1$ of SE is similar to PSCC, and is better than VPSVM. The results with CBC in *Reuters* are shown in Fig. 6g and h. From the beginning, $\mu\_F_1$ of SE is slightly less than the other two algorithms. With the increasing ratio, the performance of all algorithms first increases and then decreases. Obviously, $\mu\_F_1$ of PSCC and VPSVM approximate to zero in the ratio of 80%.

In summary, SE outperforms PSCC and VPSVM in 11 out of 13 datasets (text datasets with different classifier can be seen as different datasets). The reason is that PSCC and VPSVM can not accurately separate support vectors from the whole training instances. Thus, a lot of non-support vectors are still remained in the reduced subset, while a part of support vectors are incorrectly deleted. On the contrary, SE can effectively extract non-support vectors, which maximizes the retention of support vectors in the reconstructed subset. In addition, we can find out that non-support vector not only increases the computation cost of learning but also degrades the performance of classification, since the performance of SE in the ratio of 80% is higher than that in the original training set as shown in *Heart*.

In the second group of experiments, SE is compared with LFSVM, NNSVM, CNN and KMSVM in each dataset. Since LFSVM, NNSVM, CNN and KMSVM can not adjust the ratio, we respectively adjust the parameter of SE in order to obtain the approximate ratio of the compared algorithm. The results of these algorithms compared with SE are shown in Table 2–5, where the optimal $m\_F_1$ and $\mu\_F_1$ are highlighted in bold for each dataset.

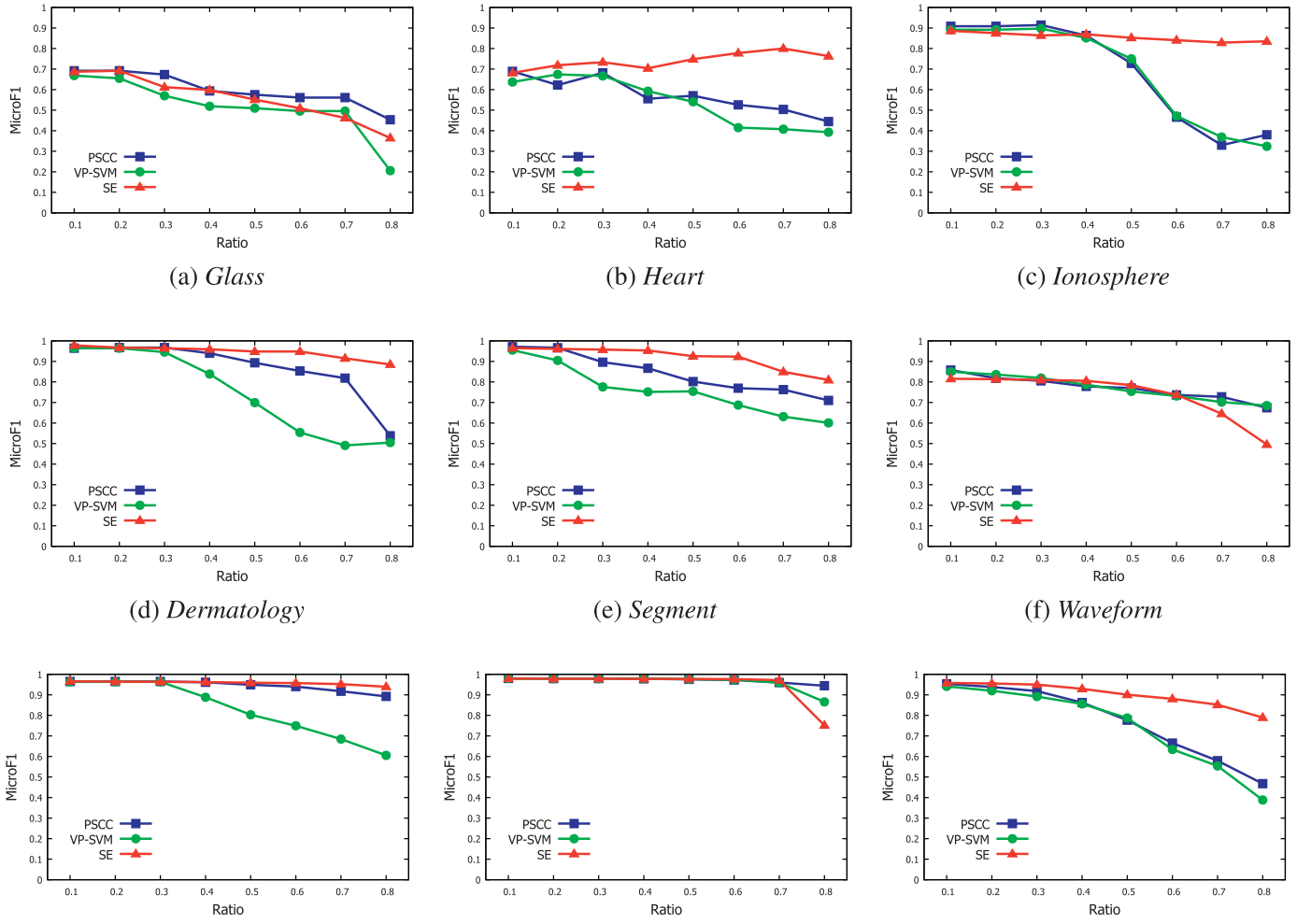



**Fig. 4.** The $\mu\_F_1$ comparison of PSCC, VPSVM and SE in nine low dimensional datasets.

**Table 3**
The comparison results of KMSVM and SE in each dataset.

| Datasets | Ratio | KMSVM $m\_F_1$ | KMSVM $\mu\_F_1$ | SE $m\_F_1$ | SE $\mu\_F_1$ |
|---|---|---|---|---|---|
| *Dermatology* | 35.65% | 0.952 | 0.962 | **0.962** | **0.967** |
| *Glass* | 25.57% | **0.661** | 0.641 | 0.561 | **0.642** |
| *Heart* | 34.15% | 0.629 | 0.630 | **0.725** | **0.721** |
| *Ionosphere* | 29.76% | **0.871** | **0.886** | 0.868 | 0.884 |
| *Isolet* | 0.12% | 0.968 | **0.967** | **0.978** | 0.963 |
| *Letter* | 3.46% | 0.952 | 0.951 | **0.963** | **0.962** |
| *Segment* | 27.93% | 0.829 | 0.846 | **0.971** | **0.978** |
| *USPS* | 0.11% | 0.977 | 0.979 | **0.982** | **0.982** |
| *Waveform* | 21.84% | **0.860** | **0.860** | 0.809 | 0.812 |
| *Newsgroup*-SVM | 6.54% | 0.933 | 0.932 | **0.935** | **0.934** |
| *Newsgroup*-CBC | 6.54% | 0.901 | 0.900 | **0.908** | **0.907** |
| *Reuters*-SVM | 24.82% | 0.450 | **0.731** | **0.458** | 0.730 |
| *Reuters*-CBC | 24.82% | 0.508 | 0.651 | **0.532** | **0.672** |

**Table 4**
The comparison results of CNN and SE in each dataset.

| Datasets | Ratio | CNN $m\_F_1$ | CNN $\mu\_F_1$ | SE $m\_F_1$ | SE $\mu\_F_1$ |
|---|---|---|---|---|---|
| *Dermatology* | 82.52% | **0.928** | **0.937** | 0.892 | 0.887 |
| *Glass* | 53.36% | **0.613** | **0.626** | 0.462 | 0.523 |
| *Heart* | 44.93% | 0.592 | 0.592 | **0.731** | **0.733** |
| *Ionosphere* | 79.10% | 0.666 | 0.670 | **0.793** | **0.812** |
| *Isolet* | 72.62% | **0.964** | **0.964** | 0.952 | 0.951 |
| *Letter* | 83.38% | **0.898** | **0.897** | 0.792 | 0.779 |
| *Segment* | 81.38% | **0.948** | **0.948** | 0.805 | 0.784 |
| *USPS* | 88.22% | **0.960** | **0.963** | 0.910 | 0.912 |
| *Waveform* | 59.26% | **0.852** | **0.850** | 0.849 | 0.849 |
| *Newsgroup*-SVM | 58.04% | 0.925 | 0.924 | **0.934** | **0.934** |
| *Newsgroup*-CBC | 58.04% | 0.912 | 0.911 | **0.924** | **0.923** |
| *Reuters*-SVM | 47.42% | **0.439** | **0.729** | 0.415 | 0.721 |
| *Reuters*-CBC | 47.42% | **0.521** | **0.680** | 0.494 | 0.670 |

The results of LFSVM compared with SE are shown in Table 2. SE outperforms LFSVM in $\mu\_F_1$ in 12 datasets except *Glass*. SE beats LFSVM in $m\_F_1$ in 10 datasets. Moreover, the disadvantage of LFSVM is that almost half of the samples were removed for each dataset, which results in a substantial decline in accuracy.

SE outperforms KMSVM in $\mu\_F_1$ in 9 datasets, and in $m\_F_1$ in 10 datasets, as shown in Table 3. For example, $\mu\_F_1$ of KMSVM are 9.1 and 13.2 percent lower than SE in *Heart* and *Segment*, respectively. Moreover, in term of reduction ratio, KMSVM performs well in some of the datasets, but it seems not suitable for some datasets such as *Isolate*, *Letter*, *USPS*, and *Newsgroup*, since the ratios are too small in these datasets.

Similarly, the results of CNN and NNSVM compared with SE are shown in Tables 4 and 5, respectively. It is easy to see that CNN has a high reduction ratio as more than half of the samples are removed in most of the datasets. In contrary, NNSVM can not show

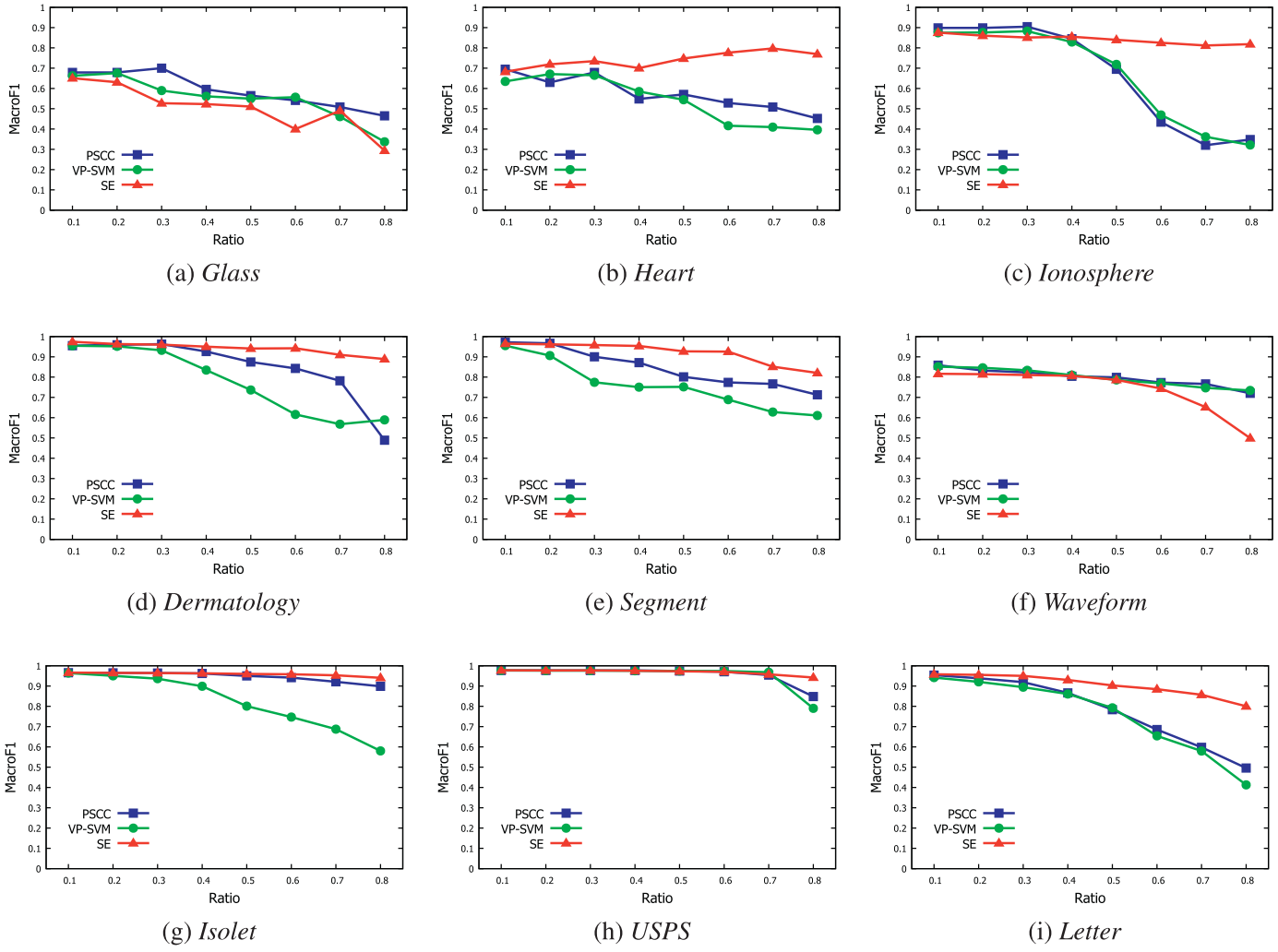



**Fig. 5.** The $m\_F_1$ comparison of PSCC, VPSVM and SE in nine low dimensional datasets.

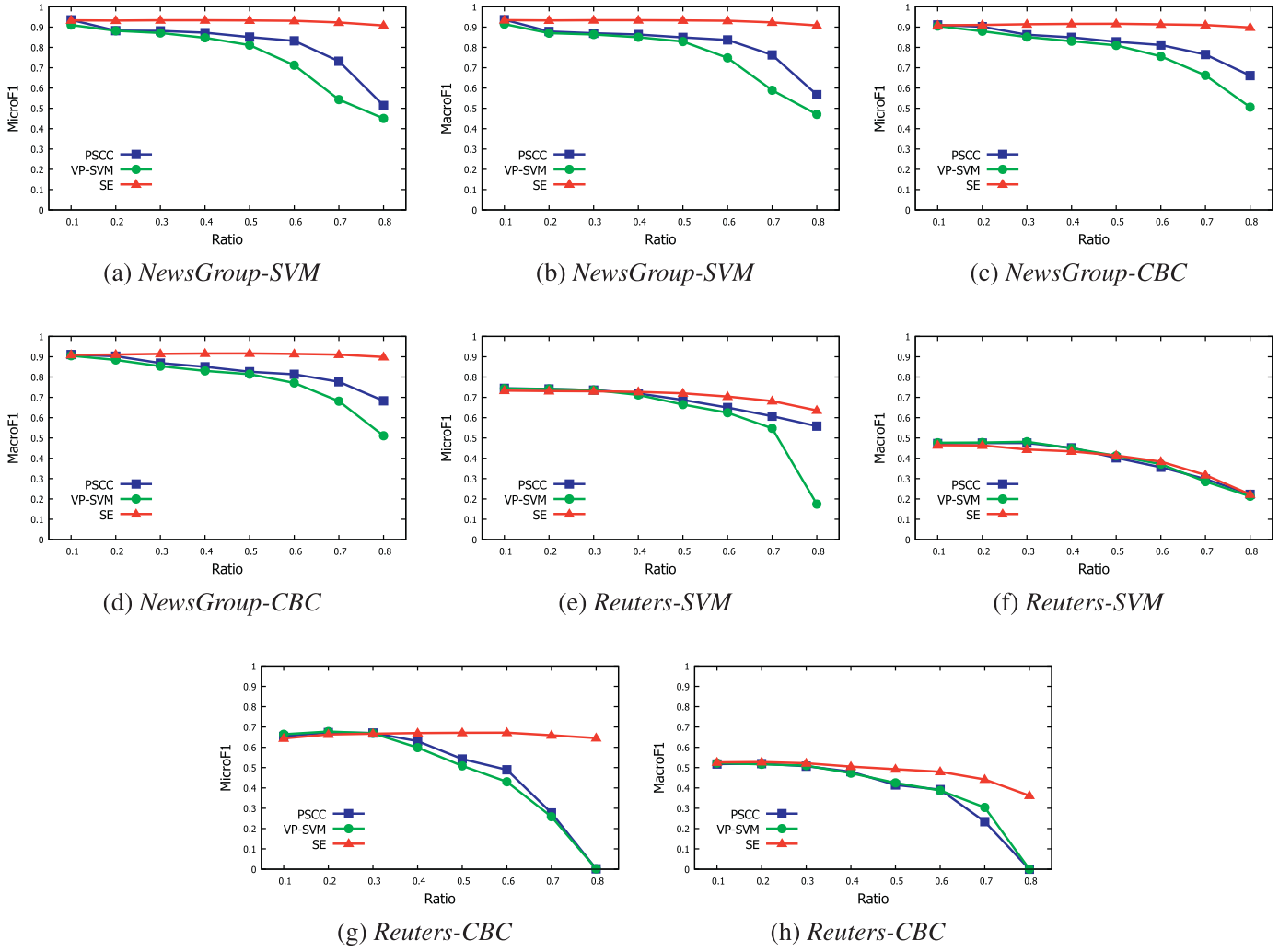
**Table 5**
The comparison results of NNSVM and SE in each dataset.

| Datasets | Ratio | NNSVM $m\_F_1$ | NNSVM $\mu\_F_1$ | SE $m\_F_1$ | SE $\mu\_F_1$ |
|---|---|---|---|---|---|
| *Dermatology* | 4.30% | 0.954 | 0.964 | **0.968** | **0.972** |
| *Glass* | 28.50% | 0.598 | **0.649** | **0.611** | 0.640 |
| *Heart* | 32.01% | **0.768** | **0.770** | 0.760 | 0.762 |
| *Ionosphere* | 11.58% | 0.882 | 0.891 | **0.885** | **0.892** |
| *Isolet* | 10.38% | 0.953 | 0.952 | **0.966** | **0.965** |
| *Letter* | 4.25% | 0.950 | 0.950 | **0.958** | **0.958** |
| *Segment* | 6.27% | **0.960** | **0.960** | 0.956 | 0.954 |
| *USPS* | 3.09% | 0.973 | 0.976 | **0.982** | **0.982** |
| *Waveform* | 22.92% | **0.864** | **0.864** | 0.861 | 0.861 |
| *Newsgroup*-SVM | 20.44% | 0.918 | 0.916 | **0.935** | **0.935** |
| *Newsgroup*-CBC | 20.44% | 0.889 | 0.888 | **0.914** | **0.914** |
| *Reuters*-SVM | 36.90% | 0.389 | 0.708 | **0.434** | **0.730** |
| *Reuters*-CBC | 36.90% | 0.365 | 0.641 | **0.519** | **0.673** |

the normal reduction ability as the ratio is less than 10% in *Isolet, Letter, Segment* and *USPS*. This observation is consistent with the previous analyses that the reduction capability of condensation strategies is comparatively higher than edition methods. In term of accuracy, CNN is better than SE, but NNSVM is worse than SE.

In order to perform a comprehensive comparison of all algorithms in each dataset, we adopt the same data analysis technique as used in [58]. Fig. 7 depicts each pair (ratio, accuracy) of algorithms in two-dimension coordinate, where the normalised Euclidean distance between each point and the ideal point (1,1) can be employed to assess the comprehensive performance of each algorithm. In this case, the "best" one is deemed as the one nearest to (1,1). With respect to the adjustable approaches, i.e. PSCC, VPSVM and SE, the points for this comparison are the best points (nearest to ideal point) chosen from the results in the first group of experiments. Regarding the approaches that can not adjust the ratio, the points come from the second group of experiments. We can easily see that SE remains the leader with six datasets (i.e., *Heart, Ionosphere, Isolet, USPS, Newsgroup* and *Reuters*), while CNN takes the crown for three datasets (i.e., *Dermatology, Letter, Segment*). Moreover, PSCC and VPSVM are the best ones in *Glass* and *Waveform*, respectively.

Specifically, the time consumption of each algorithm is shown in Table 6. Considering that the computation consumption of SE is proportional to the reduction ratio. Thus, the parameter $\xi$ is set to 0.5 in SE. Apparently, SE has a remarkable superiority in the large scale datasets, such as *Isolet, USPS, Newsgroup* and *Reuters*. For instance, SE takes only 5822 ms in *Isolet*, while the second ranked algorithm (i.e. LFSVM) consumes 34196 ms. Similarly, in *Newsgroup*, the time consumption of SE (consuming 22996 ms) is 1/80 times of VPSVM (consuming 1833814 ms) which is the close runner-up, and is 1/161 times of CNN (consuming 3709599 ms) which is the slowest algorithm. Indeed, the time consumed by CNN is far beyond the time of training SVM with the original dataset.




(a) *NewsGroup-SVM* (b) *NewsGroup-SVM* (c) *NewsGroup-CBC*

(d) *NewsGroup-CBC* (e) *Reuters-SVM* (f) *Reuters-SVM*

(g) *Reuters-CBC* (h) *Reuters-CBC*

**Fig. 6.** The $F_1$ comparison of PSCC, VPSVM and SE in *NewsGroup* and *Reuters*.

**Table 6**
The time (ms) consumption in each dataset.

| Datasets | SE | VPSVM | PSCC | LFSVM | NNSVM | KMSVM | CNN |
|---|---|---|---|---|---|---|---|
| *Dermatology* | 25 | 32 | 48 | 72 | 284 | 6895 | 109 |
| *Glass* | 31 | 15 | 16 | 31 | 98 | 140 | 94 |
| *Heart* | 32 | 16 | 16 | 47 | 109 | 187 | 124 |
| *Ionosphere* | 31 | 31 | 31 | 93 | 437 | 937 | 124 |
| *Isolet* | 5822 | 37,143 | 34,897 | 34,196 | 3,564,787 | 1,935,437 | 3,370,709 |
| *Letter* | 1007 | 921 | 858 | 1030 | 329,863 | 54,423 | 173,309 |
| *Segment* | 406 | 78 | 78 | 156 | 5320 | 5478 | 2403 |
| *USPS* | 2529 | 6459 | 5257 | 6271 | 1,892,628 | 636,411 | 619,904 |
| *Waveform* | 405 | 78 | 94 | 219 | 22,309 | 15,128 | 15,682 |
| *Newsgroup* | 22,996 | 1,833,814 | 1,875,194 | 1,910,789 | 2,665,037 | 2,867,117 | 3,709,599 |
| *Reuters* | 3672 | 361,498 | 369,112 | 361,395 | 205,982 | 246,979 | 144,133 |

Also, NNSVM has a similar disadvantage as CNN in face of large size datasets, since the time complexity is $O(N^2)$ for seeking the nearest neighbor of each instance.

In summary, we can draw the following conclusions from above experiments. In term of keeping accuracy, SE can achieve the goal of reducing the training set without degrading the classification accuracy significantly. Explicitly, SE can reduce 80% of the instances without degrading the classification accuracy in some datasets such as *Heart, Isolet, USPS*, and *Newsgroup*, and with a slight degrading in *Ionosphere, Dermatology, Segment, Letter* and *Reuters*. The performance of SE is better than PSCC and KMSVM, and far better than VPSVM, LFSVM and NNSVM. Moreover, CNN also prove itself to be the best one. In term of speed, SE has a great advantage in time consumption in face of large scale datasets. The second ranking are VPSVM, PSCC, and LFSVM due to their linear time complexity. Furthermore, CNN is proved to be the slowest one since its time complexity is approximately $O(N^3)$. In term of comprehensive performance, SE is the winner, and CNN is a very close runner-up. NNSVM is likely to be the worst one since it is always far away from the ideal point. Thus, SE offers faster and more effective training set optimization than most competitive algorithms.

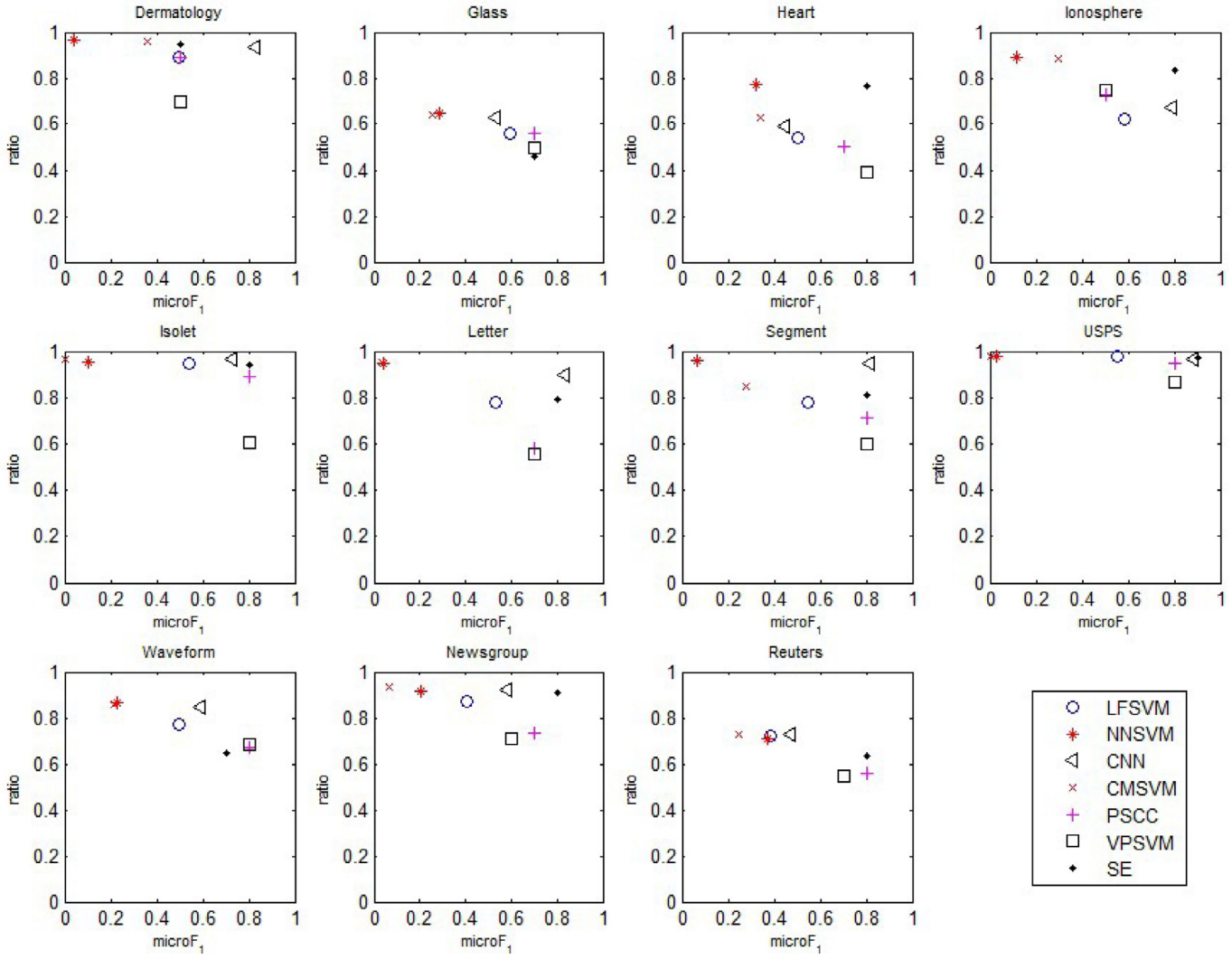


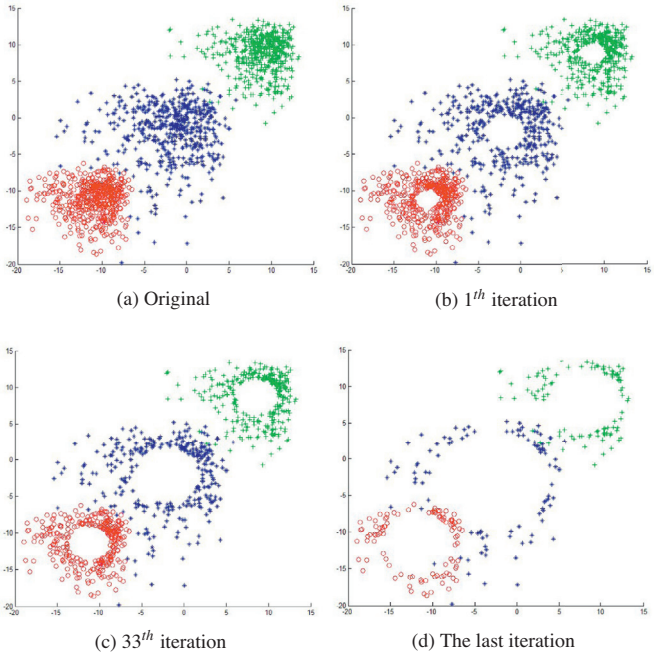

**Fig. 7.** The comprehensive comparison of all methods in each dataset.

### 5.4. Parameter analyses in shell extraction

In this section, we are devoted to illustrating the selected subsets resulting from SE. To do this, we produce a *two-dimensional* artificial dataset, which contains three classes composed of 1500 instances, based on Extreme value distribution. The complete dataset is illustrated in Fig. 8a. Fig. 8b–d show the selected subsets in the iteration process of SE, which could help to visualize and understand the way of working and the results obtained in this study. It should be appreciated that all margin points are remained but interior points are removed.

In order to deep understand how the parameters (i.e. $\lambda$ and $\delta$) impact on the performance (including accuracy and speed) of SE, we present a detailed analysis based on the artificial dataset and *Newsgroup*. Fig. 9 illustrates the selected subsets by SE with different $\lambda$ in the artificial dataset, where the two values specified in parentheses for each subgraph are respectively $m\_F_1$ and $\mu\_F_1$, which are calculated in testing the whole instances based on the SVM model trained by each selected subsets. In the beginning, as $\lambda$ is equal to 0.5, all border points are perfectly preserved as shown in Fig. 9a. However, with the increase of parameter $\lambda$, SE performs a more aggressive removal of instances in the decision boundaries as observed from Fig. 9a–d. Considering the polarization case that $\lambda$ reaches 1.5, most of the border points are removed as illustrated in Fig. 9d. Fig. 10 reveals the accuracy and iterations with the increase of parameter $\lambda$ and $\delta$ on *Newsgroup*, where $\lambda$ varies from 0.4 to 3.0 with step 0.01 as shown in Fig. 10a and b, and $\delta$ varies from 0.002 to 0.08 with step 0.002 as shown in Fig. 10c and d. With the increase of $\lambda$, the iterations gradually decline. The accuracy, however, maintains an upward trend and reaches the maximum value of 0.91 when $\lambda$ is equal to 1.0, and then declines severely. The reason is that with the increase of $\lambda$, the moving strength of RS gradually increases. Thereby, the non-support vectors far away from the original centroid point are more likely to be removed. However, if $\lambda$ exceeds the threshold, e.g. $\lambda > 1.0$, as shown in Fig. 10b, the most of support vectors will be deleted in the first iteration. Thereby, the accuracy decreases rapidly.

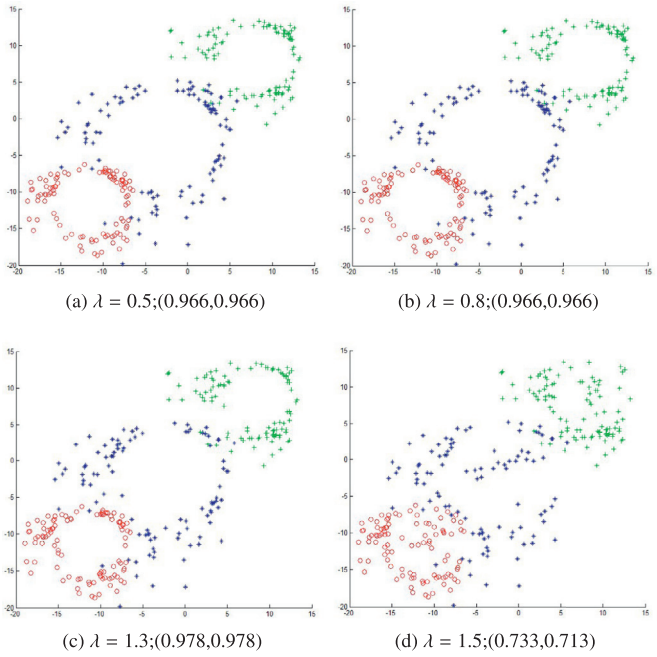
Finally, the selected subsets by different methods are shown in Fig. 11. Regarding SE, the selected subsets with different ratios (i.e., 0.3, 0.5 and 0.8) are respectively shown in Figs. 11a, b and 9b. Obviously, the optimal classification hyperplanes in the selected subsets of SE always keep fixed as in the original dataset. However, it is not true with respect to PSCC in Fig. 11d–i and VPSVM in Fig. 11j–l. In fact, in order to deal with the multi-class problem,



(a) Original (b) $1^{th}$ iteration

(c) $33^{th}$ iteration (d) The last iteration

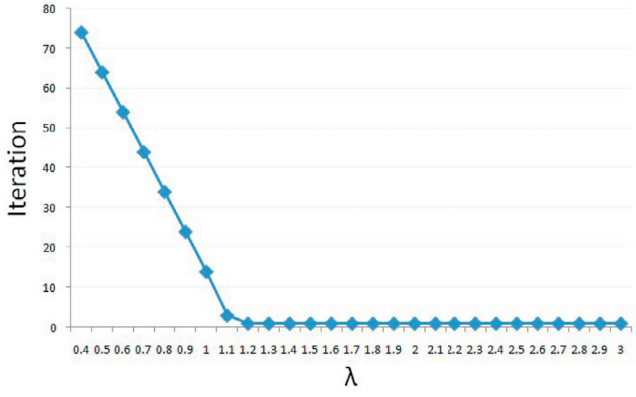
**Fig. 8.** The selected subsets in the iteration process of SE with $\lambda = 0.5$, $\delta = 0.01$ and $\xi = 0.8$.



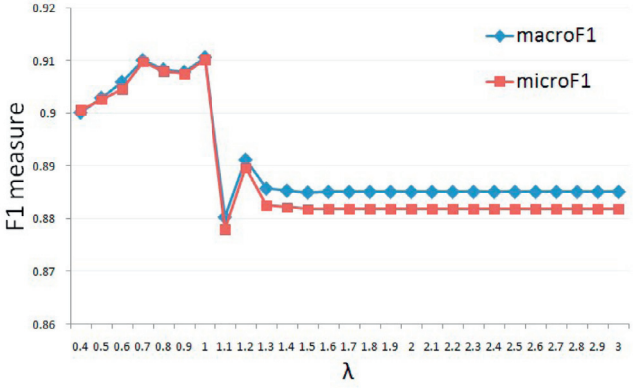
(a) $\lambda$ = 0.5;(0.966,0.966) (b) $\lambda$ = 0.8;(0.966,0.966)

(c) $\lambda$ = 1.3;(0.978,0.978) (d) $\lambda$ = 1.5;(0.733,0.713)

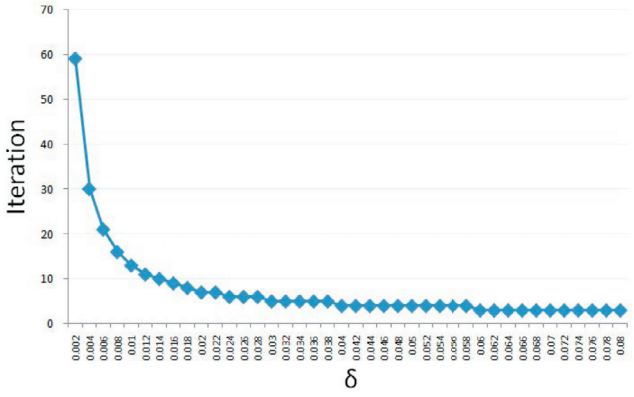
**Fig. 9.** The selected subsets by SE with different λ, when $\delta = 0.01$ and $\xi = 0.8$.



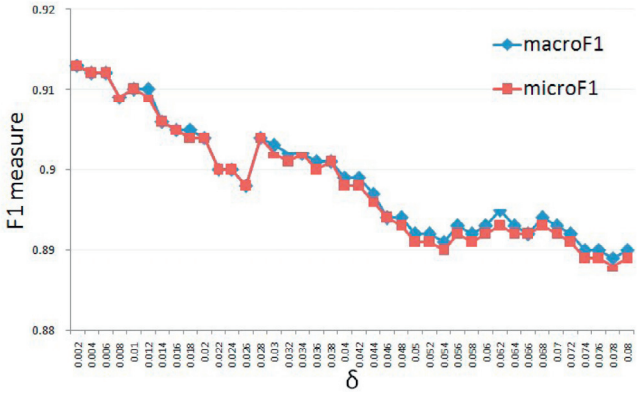

(a) The trend of iterations




(b) The trend of $F_1$

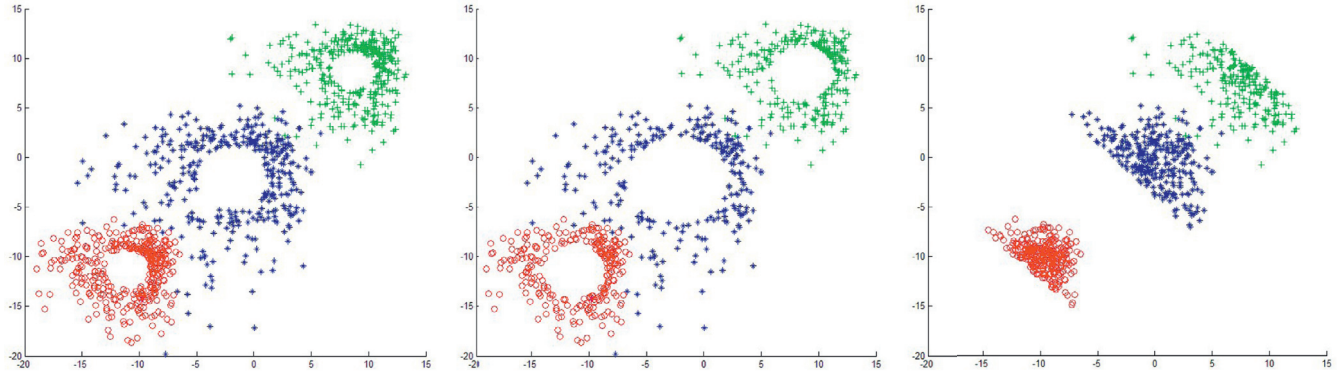



(c) The trend of iterations

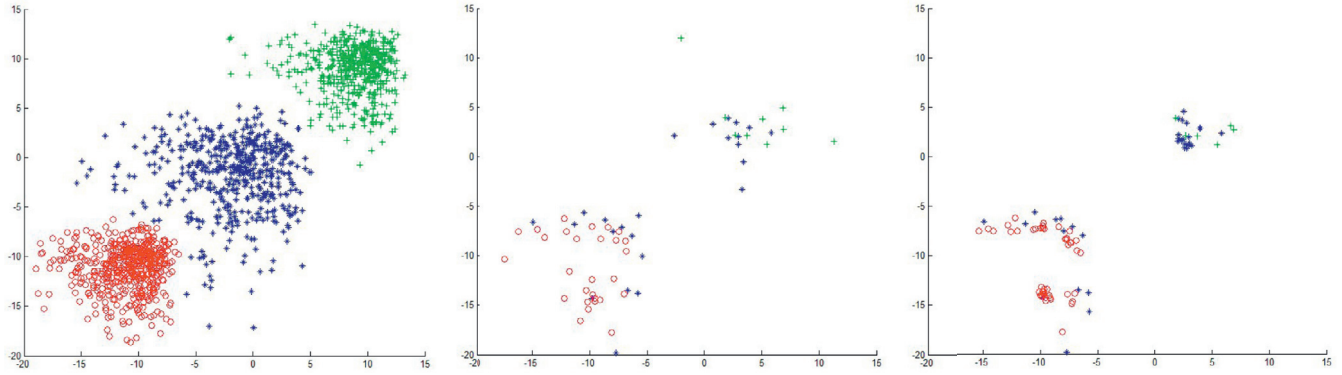



(d) The trend of $F_1$

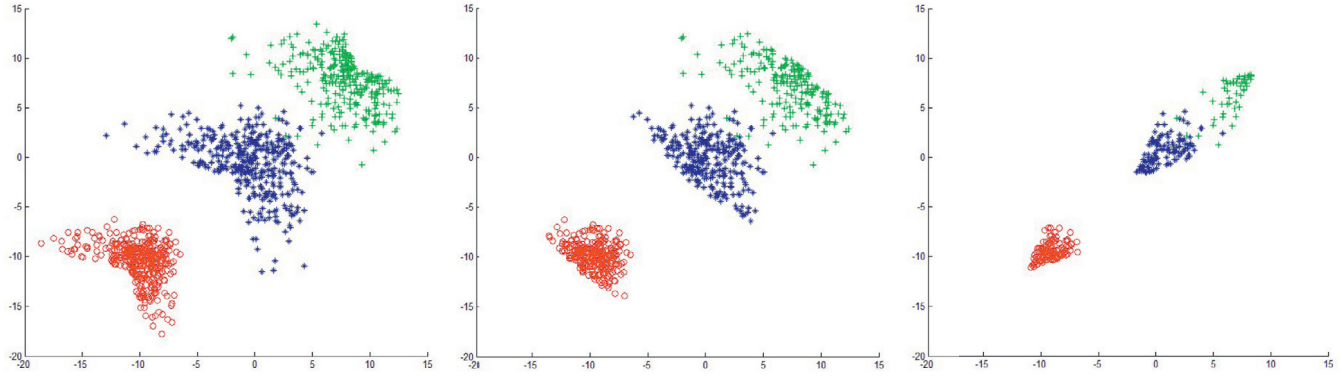
**Fig. 10.** The trend of iterations and $F_1$ with the increase of parameter λ and $\delta$ in *Newsgroup*.



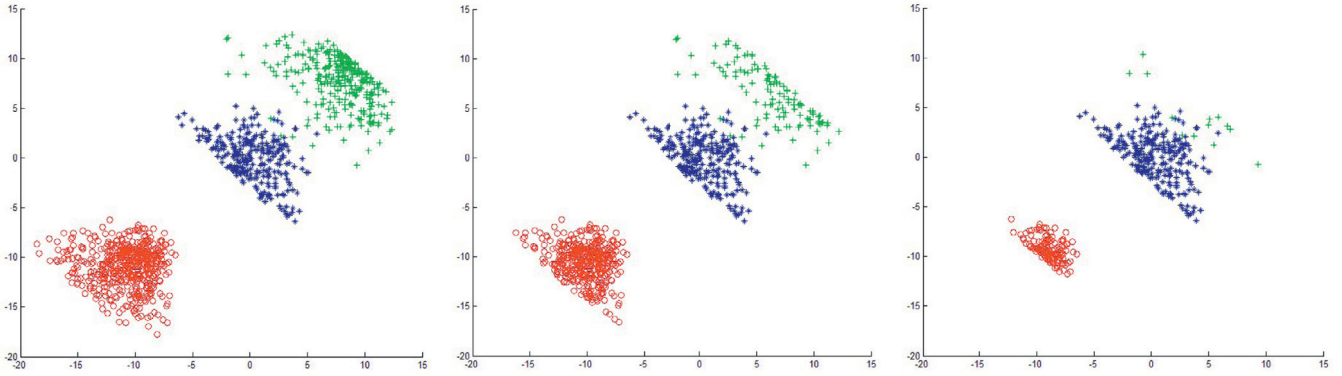
(a) SE; Ratio=0.3; (0.983, 0.983) (b) SE; Ratio=0.5; (0.969, 0.968) (c) LFSVM; Ratio=0.48; (0.956, 0.954)



(d) NNSVM; Ratio=0.01; (0.977, 0.976) (e) CNN; Ratio=0.96; (0.910, 0.909) (f) KMSVM; Ratio=0.94; (0.889, 0.875)



(g) PSCC; Ratio=0.3; (0.963, 0.962) (h) PSCC; Ratio=0.5; (0.953, 0.950) (i) PSCC; Ratio=0.8; (0.947, 0.944)



(j) VPSVM; Ratio=0.3; (0.953, 0.950) (k) VPSVM; Ratio=0.5; (0.953, 0.950) (l) VPSVM; Ratio=0.74; (0.952, 0.950)

**Fig. 11.** The selected subsets with different IS methods.

PSCC and VPSVM must transform it into a large number of binary classification problems with one class as the positive and the rest as the negative. In this case, the identification of positive border points needs the help of negative points. However, it will be difficult for PSCC and VPSVM to find out the border points if the negative classes are distributed around the positive class. Furthermore, we can also find out that CNN and KMSVM retain a small part of support vectors, while NNSVM only removes few instances overlapped with each other.

## 6. Conclusion and further study

This paper presents a new algorithm for support vector recognition to reduce training set without significantly degrading the classification accuracy of SVM. The idea of SE algorithm is an ingenious way to make use of the characteristic that the centroid point would be shifting after the uneven distributed vectors are removed off, which is totally different from the existing IS methods. Moreover, SE can be easily used in multi-class problem since it reduces a single class without the help of other classes. A large number of experiments on eleven real datasets show that SE has the advantages of flexible setting, stable performance and high efficiency. Currently, the need of fast methods for instance selection has begun to draw intensive attention among researchers. SE selects instances without requiring the whole dataset to be fitted into the memory. The advantages of SE, such as high speed and low memory consumption, indicate that it is suitable for big data processing in all fields of machine learning.

Although SE is effective, there are still some places to be improved in the future. One way is to expand SE to make it suitable for more datasets. As the precondition of SE, each class distribution should be spherical in the feature space. Thus, if the class distribution is non-convex, SE will remove some of support vectors by mistake. Another way is to build or create a new center of RS by selecting the medoids or cumuli geometric centroid (CGC) [57] of each class. Moreover, a future research line is trying to combine SE with CBC to improve the performance of classification. Recall that the centroid point of each class is not located in the geometric center of this class in the original training set. Thus, the samples that are far away from their centroid can easily be assigned incorrectly to its adjacent classes. This is the reason that CBC model has the poor performance in the original training set. However, SE can make the centroid of each class close to its geometrical center by deleting samples near the centroid constantly. Thereby, the performance of CBC should be improved in the reduced training set.

## Acknowledgements

The authors would like to thank the anonymous referees for the valuable comments and suggestions which help us to improve this paper. This work has been supported by MoE-CMCC (Ministry of Education of China - China Mobile Communications Corporation) Joint Science Fund under grant MCM20130661.

## References

[1] J.-x. Dong, A. Krzyzak, C.Y. Suen, Fast SVM training algorithm with decomposition on very large data sets, IEEE Trans. Pattern Anal. Mach. Intell. 27 (4) (2005) 603–618.
[2] V.N. Vapnik, V. Vapnik, Statistical Learning Theory, vol. 1, Wiley New York, 1998.
[3] M. Kawulok, J. Nalepa, Dynamically adaptive genetic algorithm to select training data for SVMs, in: Advances in Artificial Intelligence–IBERAMIA, Springer, 2014, pp. 242–254.
[4] L. Guo, S. Boukir, Fast data selection for SVM training using ensemble margin, Pattern Recognit. Lett. 51 (2015) 112–119.
[5] H.G. Jung, G. Kim, Support vector number reduction: survey and experimental evaluations, Intell. Transp. Syst. IEEE Trans. 15 (2) (2014) 463–476.
[6] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, K.R.K. Murthy, Improvements to platt's SMO algorithm for SVM classifier design, Neural Comput. 13 (3) (2001) 637–649.
[7] A.J. Smola, B. Schölkopf, Sparse greedy matrix approximation for machine learning, 12, Morgan Kaufmann, 2000, pp. 63–74.
[8] Y.-J. Lee, O.L. Mangasarian, RSVM: reduced support vector machines., in: SDM, vol. 1, SIAM, 2001, pp. 325–361.
[9] J.C. Platt, 12 fast training of support vector machines using sequential minimal optimization, Adv. kKrnel Methods (1999) 185–208.
[10] S. Fine, K. Scheinberg, Efficient SVM training using low-rank kernel representations, J. Mach. Learn. Res. 2 (2002) 243–264.
[11] X. Yang, J. Lu, G. Zhang, Adaptive pruning algorithm for least squares support vector machine classifier, Soft comput 14 (7) (2010) 667–680.
[12] J. Balcázar, Y. Dai, O. Watanabe, A random sampling technique for training support vector machines, in: Algorithmic Learning Theory, Springer, 2001, pp. 119–134.
[13] M.B. De Almeida, A. de Pádua Braga, J.P. Braga, SVM-KM: speeding SVMs learning with a priori cluster selection and k-means, in: Neural Networks, 2000. Proceedings. Sixth Brazilian Symposium on, IEEE, 2000, pp. 162–167.
[14] X.-L. Xiao, L.-Y. Li, X. Zhang, CM-SVM method for improving training speed of VMC, Comput. Eng. Design 22 (2006) 3–9.
[15] N. Jankowski, M. Grochowski, Comparison of instances seletion algorithms I. algorithms survey, in: International Conference on Artificial Intelligence and Soft Computing, Springer, 2004, pp. 598–603.
[16] E. Pekalska, R.P. Duin, P. Paclik, Prototype selection for dissimilarity-based classifiers, Pattern Recognit. 39 (2) (2006) 189–208.
[17] D.R. Wilson, T.R. Martinez, Reduction techniques for instance-based learning algorithms, Mach. Learn. 38 (3) (2000) 257–286.
[18] W. Lam, C.-K. Keung, D. Liu, Discovering useful concept prototypes for classification based on filtering and abstraction, IEEE Trans. Pattern Anal. Mach. Intell. 24 (8) (2002) 1075–1090.
[19] S. Garcia, J. Derrac, J. Cano, F. Herrera, Prototype selection for nearest neighbor classification: taxonomy and empirical study, IEEE Trans Pattern Anal Mach Intell 34 (3) (2012) 417–435.
[20] J.C. Bezdek, L.I. Kuncheva, Nearest prototype classifier designs: an experimental study, Int. J. Intell. Syst. 16 (12) (2001) 1445–1473.
[21] Á. Arnaiz-González, J.-F. Díez-Pastor, J.J. Rodríguez, C. García-Osorio, Instance selection of linear complexity for big data, Knowl. Based Syst. 000 (2016) 1–13.
[22] Á. Arnaiz-González, J.F. Díez-Pastor, J.J. Rodríguez, C.I. García-Osorio, Instance selection for regression by discretization, Expert Syst. Appl. 54 (2016) 340–350.
[23] M.B. Stojanović, M.M. Božić, M.M. Stanković, Z.P. Stajić, A methodology for training set instance selection using mutual information in time series prediction, Neurocomputing 141 (2014) 236–245.
[24] P. Hart, The condensed nearest neighbor rule, IEEE Trans. Inf. Theory 14 (3) (1968) 515–516.
[25] J. Ullmann, Automatic selection of reference data for use in a nearest-neighbor method of pattern classification (corresp.), IEEE Trans. Inf. Theory 20 (4) (1974) 541–543.
[26] I. Tomek, Two modifications of CNN, IEEE Trans. Systems Man Cybern. 6 (1976) 769–772.
[27] K.C. Gowda, G. Krishna, The condensed nearest neighbor rule using the concept of mutual nearest neighborhood, IEEE Trans. Inf. Theory 25 (4) (1979) 488–490.
[28] V.S. Devi, M.N. Murty, An incremental prototype set building technique, Pattern Recognit. 35 (2) (2002) 505–513.
[29] F. Chang, C.-C. Lin, C.-J. Lu, Adaptive prototype learning algorithms: theoretical and experimental studies, J. Mach. Learn. Res. 7 (10) (2006) 2125–2148.
[30] F. Angiulli, Fast nearest neighbor condensation for large data sets classification, IEEE Trans. Knowl. Data Eng. 19 (11) (2007) 1450–1464.
[31] J.A. Olvera-López, J.A. Carrasco-Ochoa, J.F. Martínez-Trinidad, A new fast prototype selection method based on clustering, Pattern Anal. Appl. 13 (2) (2010) 131–141.
[32] B.V. Dasarathy, Minimal consistent set (MCS) identification for optimal nearest neighbor decision systems design, IEEE Trans. Syst. Man Cybern. 24 (3) (1994) 511–517.
[33] G. Ritter, H. Woodruff, S. Lowry, T. Isenhour, An algorithm for a selective nearest neighbor decision rule, IEEE Trans. Inf. Theory 21 (6) (1975) 665–669.
[34] R. Barandela, F.J. Ferri, J.S. Sánchez, Decision boundary preserving prototype selection for nearest neighbor classification, Int. J. Pattern Recognit. Artif. Intell. 19 (06) (2005) 787–806.
[35] J.C. Riquelme, J.S. Aguilar-Ruiz, M. Toro, Finding representative patterns with ordered projections, Pattern Recognit. 36 (4) (2003) 1009–1018.
[36] Y. Wu, K. Ianakiev, V. Govindaraju, Improved k-nearest neighbor classification, Pattern Recognit. 35 (10) (2002) 2311–2318.
[37] H.A. Fayed, A.F. Atiya, A novel template reduction approach for the-nearest neighbor method, IEEE Trans. Neural Netw. 20 (5) (2009) 890–896.
[38] H. Shin, S. Cho, Neighborhood property–based pattern selection for support vector machines, Neural Comput. 19 (3) (2007) 816–855.
[39] H.-L. Li, C. Wang, B. Yuan, An improved SVM: NN-SVM, Chin. J. Comput. Chinese edition 26 (8) (2003) 1015–1020.
[40] E.M. Ferragut, J. Laska, Randomized sampling for large data applications of SVM, in: Machine Learning and Applications (ICMLA), 2012 11th International Conference on, vol. 1, IEEE, 2012, pp. 350–355.
[41] A. Lopez-Chau, L.L. Garcia, J. Cervantes, X. Li, W. Yu, Data selection using decision tree for SVM classification, in: Tools with Artificial Intelligence (ICTAI), 2012 IEEE 24th International Conference on, vol. 1, IEEE, 2012, pp. 742–749.

[42] A. Lyhyaoui, M. Martinez, I. Mora, M. Vaquez, J.-L. Sancho, A.R. Figueiras-Vidal, Sample selection via clustering to construct support vector-like classifiers, Neural Netw. IEEE Trans. 10 (6) (1999) 1474–1481.
[43] J. Chen, C. Zhang, X. Xue, C.-L. Liu, Fast instance selection for speeding up support vector machines, Knowl. Based Syst. 45 (2013) 1–7.
[44] R. Koggalage, S. Halgamuge, Reducing the number of training samples for fast support vector machine classification, Neural Inf. Process. Lett. Rev. 2 (3) (2004) 57–65.
[45] C.-F. Tsai, K.-C. Cheng, Simple instance selection for bankruptcy prediction, Knowl. Based Syst. 27 (2012) 333–342.
[46] X. Li, W. Yu, Fast support vector machine classification for large data sets, Int. J. Comput. Intell. Syst. 7 (2) (2014) 197–212.
[47] F. Chang, C.-Y. Guo, X.-R. Lin, C.-J. Lu, Tree decomposition for large-scale SVm problems, J. Mach. Learn. Res. 11 (2010) 2935–2972.
[48] S.L. Lam, D.L. Lee, Feature reduction for neural network based text categorization, in: Database Systems for Advanced Applications, 1999. Proceedings., 6th International Conference on, IEEE, 1999, pp. 195–202.
[49] J.R. Cano, F. Herrera, M. Lozano, Using evolutionary algorithms as instance selection for data reduction in KDD: an experimental study, Evol. Comput. IEEE Trans. 7 (6) (2003) 561–575.
[50] S. Garcı, I. Triguero, C.J. Carmona, F. Herrera, et al., Evolutionary-based selection of generalized instances for imbalanced classification, Knowl. Based Syst. 25 (1) (2012) 3–12.
[51] C. Shujuan, L. Xiaomao, Z. Jun, et al., Fuzzy support vector machine of dismissing margin based on the method of class-center, Comput. Eng. Appl. 42 (22) (2006) 146–149.
[52] L. Yu, Y. Wende, H. Dake, L. Yu, Fast reduction for large-scale training data set [j], J. Southwest Jiaotong Univ. 4 (8) (2007) 15–22.
[53] L.Q.J.L.C. ZHOU, W. Da, Pre-extracting support vector for support vector machine based on vector projection [j], Chin. J. Comput. 2 (32) (2005) 12–20.
[54] H.P. Graf, E. Cosatto, L. Bottou, I. Dourdanovic, V. Vapnik, Parallel support vector machines: the cascade SVM, in: Advances in Neural Information ProcessingSystems, 2004, pp. 521–528.
[55] S. Wang, Z. Li, C. Liu, X. Zhang, H. Zhang, Training data reduction to speed up SVM training, Appl. Intell. 41 (2) (2014) 405–420.
[56] J. Friedman, T. Hastie, R. Tibshirani, The Elements of Statistical Learning, Springer Series in Statistics, vol. 1, Springer, Berlin, 2001.
[57] T.T. Nguyen, K. Chang, S.C. Hui, Supervised term weighting centroid-based classifiers for text categorization, Knowl. Inf. Syst. 35 (1) (2013) 61–85.
[58] M.T. Lozano, J.S. Sánchez, F. Pla, Using the geometrical distribution of prototypes for training set condensing, in: Current Topics in Artificial Intelligence, Springer, 2004, pp. 618–627.