Very large-scale data classification based on K-means clustering and multi-kernel SVM

Article *in* Soft Computing · June 2019

DOI: 10.1007/s00500-018-3041-0

Jake Luo
University of Wisconsin - Milwaukee

Soft Computing
https://doi.org/10.1007/s00500-018-3041-0

METHODOLOGIES AND APPLICATION

Very large-scale data classification based on K-means clustering and multi-kernel SVM

Tinglong Tang[1,3] · Shengyong Chen[1,2] · Meng Zhao[2] · Wei Huang[2] · Jake Luo[4]

Abstract

When classifying very large-scale data sets, there are two major challenges: the first challenge is that it is time-consuming and laborious to label sufficient amount of training samples; the second challenge is that it is difficult to train a model in a time-efficient and high-accuracy manner. This is due to the fact that to create a high-accuracy model, normally it is required to generate a large and representative training set. A large training set may also require significantly more training time. There is a trade-off between the speed and accuracy when performing classification training, especially for large-scale data sets. To address this problem, a novel strategy of large-scale data classification is proposed by combining K-means clustering technology and multi-kernel support vector machine method. First, the K-means clustering method is used on a small portion of the original data set. The clustering stage is designed with a special strategy to select representative training instances. Such method reduces the needs of creating a large training set as well as the subsequent manual labeling work. K-means clustering method has two characteristics: (1) the result is greatly influenced by the cluster number k, and (2) the optimal result is difficult to achieve. In the proposed special strategy, the two characteristics are utilized to find the most representative instances by defining a relaxed cluster number *k* and doing K-means repeatedly. In each K-means clustering step, both the nearest and the farthest instance to each cluster center are selected into a set. Using this method, the selected instances will have a representative distribution of the original whole data set and reduce the need of labeling the original data set. An outlier detection method is applied to further delete the outlier instances according to their outlier scores. Finally, a multi-kernel SVM is trained using the selected instances and a classifier model can be obtained to predict subsequent new instances. The evaluation results show that the proposed instance selection method significantly reduces the size of training data sets as well as training time; in the meanwhile, it maintains a relatively good accuracy performance.

Keywords Very large-scale classification · Multi-kernel SVM · Outlier detection · K-means clustering

1 Introduction

When developing a classification model for very large data sets with a high dimension of features, we usually face the challenges of the speed and accuracy. To address the problems created by high-dimensional features, a common method is to reduce the feature dimension by eliminating or compressing the features. This type of technique is called feature reduction methods. To address the problem of processing a very large number of training data, a popular method is to reduce the size of the training set by selecting a representative subset. This type of classification technique is called data reduction technologies (DRTs). In this paper, we mainly focus on developing novel algorithms to select representative training samples among very large data sets. We aim to address two challenges. The first challenge is that it is time-consuming to label a large number of training instances. The second challenge is that it is difficult to achieve high training efficiency and accuracy simultaneously when using very large training sets. Normally to train a high-accuracy model, it would require us to use as many training samples as possible. When dealing with large-scale data, this could lead to

Communicated by V. Loia.

✉ Shengyong Chen
sy@ieee.org

[1] Zhejiang University of Technology, Hangzhou 310032, China

[2] Tianjin University of Technology, Tianjin 300384, China

[3] China Three Gorges University, Yichang 443002, China

[4] University of Wisconsin-Milwaukee, Milwaukee, WI 53211, USA

Published online: 29 January 2018

a significant increase of training time or even intractable or failed training sessions. Therefore, it is a significant challenge to classify very large-scale data sets accurately and rapidly simultaneously without the need of labeling a large number of instances.

Data reduction technologies (DRTs) are a type of a method to address the challenge of very large training data set. DRTs can reduce the computer memory required to store the data; hence, it accelerates the classification algorithms (Dornaika and Aldine 2015). There are mainly three types of reduction technologies (Wilson and Martinez 2000): instance selection (IS) (Silva et al. 2016), prototype generation (PG) (Triguero et al. 2012; Rezaei and Nezamabadi-Pour 2015) and prototype selection (PS) (Valero-Mas et al. 2016). The prototype generation approach replaces the original instances with new artificial ones; instance selection and prototype selection methods attempt to find a representative subset of the initial training set while trying to maintain the predictive power.

Khosravani et al. (2016) proposed a randomized approximation convex hull algorithm which can be used for high-dimensional data in an acceptable execution time with a low memory requirement. Though the execution time is acceptable for high-dimensional data, the speed is slow when processing very large-scale data sets because of the inevitable convex hull computation. The work proposed by Silva et al. (2016) provided an extension to the Markov geometric diffusion method for instance selection. They used the diffusion process to capture geometric characteristics from the data. Through inferences on this information, the algorithm can determine the representativeness of each instance as well as the instance's contribution to the data. However, this method focused on preserving information rather than improving the classification accuracy and cannot obtain an optimal result with a fast speed and good accuracy. Lin et al. (2015) introduced the Representative Data Detection approach based on outlier pattern analysis and prediction. They used instance-based learning method 3 (IB3), decremental reduction optimization procedure 3 (DROP3) and genetic algorithms (GA) to generate representative data sets (RD) and unrepresentative data sets (URD) and then used backpropagation neural network (BPNN), k-nearest neighbor (k-NN), classification and regression tree (CART) decision tree, naive Bayes (NB) and support vector machine (SVM) to construct the classifier for RD/URD detectors. Zhai et al. (2016) employed the mapper method of MapReduce (Dean and Ghemawat 2008) to partition large data sets into small subsets and obtained representative subsets based on a voting mechanism. Arnaiz-González et al. (2016) adopted a locality-sensitive hashing method to find similarities between instances. For very large-scale data sets, their methods will need several to hundreds of hours' execution time.

Instance selection can also be achieved by using the data sampling method, such as random sampling, stratified sampling and cluster sampling. These sampling technologies are studied in Hamidzadeh et al. (2016), Neugebauer et al. (2016), Onan (2015), Stojanović et al. (2014), Cavalcanti et al. (2013) and Sun and Li (2011). Works based on clustering methods to select instance are categorized as prototype selection by clustering (Olvera-López et al. 2010). Chen et al. (2016) proposed a gene selection method based on clustering, in which dissimilarity measures were obtained through kernel functions. The adaptive distance was used to learn the weights of genes during the clustering process. Whelan et al. (2010) also used the K-means clustering method to reduce the data. The cluster centers, together with those closest to the centers, are chosen as representatives for data instances.

In this work, we also use the K-means algorithm as a foundation for sampling training data. We try to select representative instances from a large-scale unlabeled data set by designing a simple yet effective strategy to reduce the labeling work and training time. This strategy is very efficient when combined with multi-kernel support vector machine (SVM). Firstly, we use the K-means method to cluster a portion of the data set and choose the nearest and farthest instances in corresponding to each of the cluster centers. Since K-means does not always converge to the global optimum and could produce a different clustering result in each turn, we leverage this characteristic as a mechanism for training sample generation. The K-means clustering algorithm is repeatedly run several times in order to obtain multiple cluster centers, and the nearest and farthest instances are collected as the initial sample data. The instances in the initial set can represent the distribution of the original whole data set. We then define these selected instances as the representative instances. However, these farthest instances of each center could be outlier instances. We use an outlier detection method to detect and delete the most abnormal outliers. The remaining selected data instances will be presented to an expert for labeling. After the instances are labeled, they are used as the training set to train an SVM classier model. Since the multi-kernel method showed a better performance in prior studies, we choose to train a multi-kernel SVM to enhance the performance. The proposed method has two major contributions:

1. A novel algorithm is proposed to select representative instances for multi-kernel learning. The method combines the advantages of the K-means clustering method and the outlier detection method to find the representative samples in a large-scale data set and the multi-kernel learning method to train an accuracy classifier. The selected training instances provide a good representation to the distribution of the original very large-scale data set, which is not only simple to implement but also efficient for training section.

2. The advantage of nonsupervised method (i.e., K-means clustering, outlier detection) and supervised method (i.e., multi-kernel SVM) is combined in this work, which produces satisfactory classification outcomes when using large-scale data sets.

The rest of the paper is structured as follows. Section 2 reviews the related works of multi-kernel SVM and K-means methods. Section 3 presents the details of the proposed method. Experiments and evaluation results are reported in Sect. 4. Finally, Sect. 5 concludes and summarizes the work.

2 Related works

2.1 K-means

K-means is an unsupervised learning method. It takes an input parameter k and partitions a target data set into k numbers of clusters, such that the inter-cluster similarity is low. The algorithm aims at minimizing the following objective function:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \tag{1}$$

where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point and its cluster center. Therefore, the K-means algorithm can be described as following steps:

1. Define a target cluster number k.
2. Randomly select k data points as the initial cluster centers.
3. Repeat until the mean values of clusters do not change anymore:
   {Each data point is assigned to the most similar cluster according to the distance to the centers;
   Update the mean value of each cluster and calculate new centers;
   }

The K-means clustering algorithm has two known characteristics: one is that it needs a predefined cluster number k as a prerequisite parameter for clustering; however, normally, we do not know the best number of classes a data set can produce without prior knowledge. The other characteristic is that every time we run K-means, it could result in generating a different set of clusters due to the random selection of initial centers of the algorithm.

2.2 Multi-kernel SVM

Let $x_1, \ldots x_n \in R^d$ be a set of n training samples. Let y_i be the class label of x_i, and its value is $+1$ and -1. Let $\phi(\cdot) : R^d \rightarrow H$ be a probably nonlinear mapping from R^d to a higher-dimensional feature space H. A kernel function of x_i and x_j is defined as the inner product between $\phi(x_i)$ and $\phi(x_j)$. It is expressed as $K(x_i, x_j = \phi(x_i)^{\mathrm{T}}\phi(x_j))$, where T denotes the transpose of a vector.

Solving the standard SVM is to minimize the following objective function in feature space:

$$\begin{aligned} &\min_{w,b,\xi} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i \\ &\text{s.t.} \quad y_i(w^{\mathrm{T}}\phi(x_i) + b) \geq 1 - \xi_i \\ &\qquad \xi_i \geq 0, i = 1, \ldots, l \end{aligned} \tag{2}$$

where ξ is the vector of slack variables and C is the regularization parameter used to impose a trade-off between the training error and generalization.

Using a set of n base kernels instead of a single kernel K, the multi-kernel objective function is reformulated as follows:

$$\begin{aligned} &\min_{w,b,\xi} \frac{1}{2}\sum_{k=1}^{n}\mu_k||w_k||^2 + C\sum_{i=1}^{n}\xi_i \\ &s.t. y_i\left(\sum_{k=1}^{n}\mu_k w_k^{\mathrm{T}}\phi_k(x_i) + b\right) \geq 1 - \xi_i \\ &\qquad \xi_i \geq 0, \quad i = 1, \ldots, l \\ &\qquad \sum_{k=1}^{n}\mu_k = 1 \\ &\qquad \mu_k \geq 0, \quad k = 1, \ldots, n \end{aligned} \tag{3}$$

SimpleMKL (Rakotomamonjy et al. 2008) is a fast method to solve the multi-kernel learning problem. It has been used in many other works (Liu et al. 2012, 2015; Wu et al. 2015). We also directly incorporate their method into our work.

3 Proposed method

To achieve a good accuracy as well as a fast training speed, we propose a three-stage algorithm. In the first stage, the clustering method of K-means is applied on a small portion of the original whole data set. The ratio of the selected instances has effects on their representativeness and the speed of the whole algorithm: larger ratio of them will get better representativeness but result in more clustering time. In our method, we choose the ratio value as small as possible when the result is including enough different classes of samples for a data set. For most of the data sets, 10% of the original data will be enough for clustering in our experiments. The K-means clustering processing is designed with a special strategy by defining a relaxed clustering number k and repeatedly running K-means several times to find the representative instances as an initial training set. The parameter of repetitional times is defined as RT in our method. The initial set is further filtered according to samples' outlier scores calculated by an outlier detection method. The resulting training samples are presented to an expert for labeling. After outlier filtering, the selected training set has a much small size, yet the selected samples have a sufficient representation of the original data. Finally, a multi-kernel SVM classifier is trained based on the selected set. An evaluation is conducted using the rest samples of the original large data set.

3.1 Training instances selection stage

The results of K-means are greatly influenced by the predefined cluster number k and the chosen clustering starting point; therefore, it is possible to obtain different clusters when we run multiple K-means sections on the same data set. In our algorithm, we leverage this characteristic of K-means to find a smaller set of instances that represents the distribution of the whole data set. Labeling these selected instances for training will require much less human work. We define a relaxed cluster number and run K-means repeatedly, so that we will get different centers and margin points at each time. These points will represent the whole data set well enough for training a high-accuracy model. Therefore, in the first stage of our algorithm, the clustering number of K-means is defined as an integer around 5–30, and the K-means algorithm will be repeatedly run 5–30 times. The nearest and farthest sample instances to each cluster center will be collected. In here, we utilize the characteristics of K-means that it converges to different clusters at each run to find representative instances. Such characteristics enable us to repeatedly collect the farthest and nearest instances to their centers at every round of clustering. The selected instances are treated as the representative training instances.

3.2 Outlier detection and reduction stage

First, the repeating data instances in the collected data set are directly deleted to remove instance duplication. Some outlier instances still could have a negative impact to build a classifier. We use an outlier detection method proposed in Kim (2013) to examine the outlier score of each instance.

To find the outlier samples, we compute the Kolmogorov–Smirnov statistic between a given point j to another point in the collected set using

$$\mathrm{KS}(p_j - p_i) = \sup_x \left| F_{p_j}(x) - F_{p_i}(x) \right| \tag{4}$$

where F_{p_j} is the distances from point j to another point in the collected set.

The average of the Kolmogorov–Smirnov test statistics will be used to compute the KSE test statistic, which will generate the outlier score based on formula:

$$\mathrm{KSE}(p_j) = \frac{1}{n-1} \sum_{i=1}^{n} \mathrm{KS}(p_j - p_i) \tag{5}$$

If the average KS statistic, called KSE statistic here, is bigger than a threshold for a tested instance, the corresponding instance will be treated as an outlier and deleted directly. It is difficult to find a suitable threshold. So we simply delete the instances with the biggest outlier score and repeat this process several times. The repeat time is defined as repeating outlier detection time (ROT) in our method. After that, we get the reduced training set R which will be labeled by a human expert for training.

3.3 Multi-kernel SVM training and testing stage

After being labeled, the selected data set is used as the training set for building a classifier in the third stage.

It has been proved that the multi-kernel SVM has a better performance than traditional SVMs. However, it is slow to train a multi-kernel SVM when using a large-scale data set. After the clustering and data reduction stages of our method, the size of the selected instances is greatly reduced; therefore, it is more suitable for multi-kernel training. We use the existing method of SimpleMKL method proposed in Rakotomamonjy et al. (2008) to train a model because the SimpleMKL implementation converges faster and more efficient when compared with other multi-kernel learning methods.

3.4 Algorithm description

Our algorithm is described in Table 1.

4 Experiments

In this section, experiments are conducted using data sets extracted from the UCI machine learning repository (Lichman 2013) and KEEL data set repository (Alcalá-Fdez et al. 2011). All experiments were implemented using the MATLAB R2015b version running on an Intel(R) Core i5-3470 CPU @3.20 GHz 3.20 GHz, 4.0 GB RAM machine.

In the experiments, we mainly evaluate the accuracy and execution time. The accuracy is defined as the percentage of correctly classified instances over the number of all testing instances. The execution time is defined as all the CPU time of each method.

Table 1 Proposed algorithm

```
Input: large-scale unlabeled data
Output: predicted label

Step 1 Clustering stage
1.1 select a small percentage of instances;
1.2 set the parameter of target cluster number of K-means as k=5 to 30;
1.3 set the clustering repeating sessions as RT=5 to 30;
1.4
    for i=1:RT
        use the K-means clustering method to cluster data into k classes;
        select the nearest and farthest instances to each cluster center
            and add them to the initial training set;
    end

Step 2 Outlier detection and reduction stage
2.1 delete repetitive instances in the initial training set;
2.2 compute the outlier scores of all instances;
2.3
    for i=1:ROT
        delete the instances with the highest score;
    end
    get the reduced training set;

2.4 label the reduced set;

Step 3 Training stage
3.1 set multi-kernel SVM parameters;
3.2 train the multi-kernel SVM classifier based on labeled training set;

Step 4 Evaluation
4.1 use the rest of instances to predict and test;
4.2 evaluate the performance.
```

4.1 Performance comparison on small- to large-sized data sets

4.1.1 Experimental setup

In this section, we retrieve four data sets from small to large scales: Wisconsin breast cancer (breast-w), car evaluation (car) and Diabetic Retinopathy Debrecen Data Set (messidor) are from UCI machine learning repository; spambase data set is from the KEEL data sets repository. We compare our method with LibSVM (Chang and Lin 2011).

Conventional SVM (i.e., LibSVM) needs a large number of training instances to obtain a good accuracy, but our method only needs a small set of instances at the beginning. To achieve the best possible result of the compared method, the data sets are firstly divided into a training set with 80% of the data for LibSVM classifier training and a testing set with 20% of the data for testing. In our algorithm, we only need at most 10% of the data sets to conduct K-means clustering for instance selecting and define different k value to specify the clustering number for different data sets; K-means will do RT times and the farthest and nearest instances of each cluster center will be selected; outlier samples of the selected set are also deleted according to the outlier scores and the deleted set will be used to train the multi-kernel SVM classifier; the rest instances of each data set are used for testing.

Table 2 Accuracy based on small- to large-sized data sets

Data set	Samples	Training instance size		Accuracy (%)		Running time (s)	
		Proposed	LibSVM	Proposed	LibSVM	Proposed	LibSVM
Breast-w	683*10	**19**	546	92.44	**92.75**	1.61	**0.17**
Messidor	1151*20	**36**	920	60.23	**64.22**	2.22	**0.56**
Car	1728*6	**56**	1282	87.35	**90.20**	3.67	**0.26**
Spambase	4601*57	**26**	3680	**77.75**	77.22	**9.39**	12.30

Bold text highlights the best results of comparisons

4.1.2 Performance comparison

The accuracy and execution time comparisons based on small- to large-sized data sets are shown in Table 2.

Conventional LibSVM needs a large number of instances for training, which means that it will cost much labor and time to label the instances. However, in our method, we only need to label a small selected portion of representative instances for training without accuracy declining. Thus, our sample selection method significantly reduces the laborious work of human manual labeling. It is shown in columns 5 and 6 of Table 2 that our method can achieve a similar accuracy when compared with LibSVM even using very few representative instances for training on different scales of data sets.

The computational cost of K-means is $O(Tkn)$ where T is the number of iterations, k is the clustering number, and n is the number of objects in the input data set (Huang 1998). The computational cost of solving the SVM problem has both a quadratic n^2 when C (the regularization parameter in Eq. 2) is small and a cubic component n^3 when C gets large (Bottou and Lin 2007), where n is the number of instances. So when n is very small, the speed of our method is obviously slower than traditional SVM, which can be seen in column 7. However, when the scale of data sets becomes larger, the speed of our method is significantly faster than conventional LibSVM. The results show that even on a small number of data sets, such as breast-w, messidor and car data set, our method can achieve similar accuracy and speed with conventional SVM; when the scale of data set becomes larger, such as spambase data set with 4601 instances and 57 dimensions, the proposed method is faster and more accurate than LibSVM.

The advantages of accuracy and speed of our method are very evident when classifying large-scale data sets. In the next evaluation, we will analyze the performances of our method on larger-scale data sets.

4.2 Performance analysis on very large-scale data sets

4.2.1 Experimental setup

To demonstrate the effectiveness of applying our method on very large data sets, we evaluate five very large-scale data sets: coil2000 data set is from the KEEL data sets repository; bank marketing database, skin segmentation data set, covertype data set and Localization Data for Person Activity data set (Protein Prediction) are from the UCI data repository. Firstly, we analyze the accuracy and speed when using our method on very large-scale data sets. Secondly, we make comparisons with other methods.

4.2.2 Accuracy and speed analysis

The performance of our method when classifying very large-scale data sets is analyzed in Table 3. In Table 3 column 3, parameters include the ratio of instances used for clustering (ratio), the cluster number of K-means clustering (k), the repeat times of K-means clustering (RT) and the repeat times of deleting the outliers with the highest outlier scores (ROT). After clustering and outlier detection processing, a small number of representative instances are obtained. We list the number of representative training instances in column 4. Only these instances should be labeled. We use the ground truth of the data sets as the label.

As shown in Table 3 column 3, the proposed algorithm only requires a small percentage of the original data during the K-means clustering stage. In the experiments, 2% of the coil2000 data set, 10% of the bank marketing data set, 0.05% of the skin segmentation data set and 2% of the covertype data set are selected as data sources to search for representative training instances. After the data reduction stage completed using the outlier method, the number of resulting representative training instances is even small. As shown in Table 3 column 4, the training instance size is only 50, while the original covertype data set has a number of 581,012 instances, and the representative sample set is a very small subset of the original data ($50/581{,}012 = 0.0086\%$). Therefore, the manual labeling work can be greatly reduced. The selected representative instance set also improves training speed using the multi-kernel SVM algorithm. For example, we obtain a classification accuracy of 95.91% using the 50 representative instances from the 2% of the covertype data set, and it is notable that the total execution time of our method is only 58.63 s.

Table 3 Performance on very large-scale data sets

Data set	Size	Parameters	Performance						
			Training	Accuracy	Execution time (s)				
			Size	%	Clustering	Deleting	Training	Testing	Total

Coil2000	9822*85	ratio = 0.02;
		k = 26;
		RT = 12;
		ROT = 26
			68	92.73	0.39	1.60	2.70	0.32	5.01

Bank marketing	45,211*17	ratio = 0.1;
		k = 9;
		RT = 15;
		ROT = 5
			13	88.22	2.55	0.02	0.27	0.05	2.89

Skin segmentation	245,057*4	ratio = 0.0005;
		k = 5;
		RT = 15;
		ROT = 5
			18	93.21	0.07	0.05	0.37	0.61	1.10

Covertype
(Aspen vs others)
	581,012*54	ratio = 0.02;
		k = 10;
		RT = 30;
		ROT = 5
			50	95.91	55.87	0.34	1.00	1.42	58.63

Table 4 Comparison with similar methods

Data size	Average processing time			Average accuracy (%)		
	Baseline	ReDD	Proposed	Baseline	ReDD	Proposed
145,751*74	1108.54 h	363.19 h	**299.8 s**	96.35	98.57	**99.24**

Bold text highlights the best results of comparisons

Table 5 Accuracy comparison with random-selection method

Data set	Data size	Training number	Random-selection method			Proposed method		
			Min	Max	Average	Min	Max	Average
Coil2000	9822*85	68	88.00	93.90	91.35	89.67	94.01	**92.73**
Bank marketing	45,211*17	13	59.62	88.73	83.13	88.21	88.21	**88.21**
Skin segmentation	245,057*4	18	82.21	98.22	92.47	87.06	97.45	**93.21**
Covertype	581,012*54	50	1.63	98.37	83.05	95.82	96.71	**95.91**

Bold text highlights the best results of comparisons

4.2.3 Comparisons

For very large-scale data sets, Lin et al. (2015) proposed a framework named as Representative Data Detection (ReDD) which has some similarity to ours that is firstly performing instances selection and training classifier in the next. They compared their method with the baseline methods that use GA, IB3 or DROP3 for instance selection and CART, k-NN or SVM for classification. When we compare the results that they listed in their work to ours on the evaluation data set of Localization Data for Person Activity (Protein prediction), we find that their speed is much slower and the accuracy is lower than ours, which can be seen in Table 4.

Because of the large data size, there is either not enough memory or there will cost hundreds of hours to run LibSVM or ReDD method mentioned above. In our method, after clustering and outlier detection we will obtain some representative instances for multi-kernel training. To provide a comparison to demonstrate the effectiveness of the proposed sample selection method, we use conventional random-selection method to randomly select the same number of training instances to compare with our method for training the multi-kernel SVM. The rest of data sets are used for testing according to their ground truth and calculating the accuracy. We report the results of minimum accuracy, maximum accuracy and average accuracy from the 15 rounds of evaluation and present the comparisons in Table 5.

The random training sample selection method could achieve a good accuracy occasionally in some data sets, but it is not very consistent. It is shown in Table 5 that the results of random method have a great fluctuation between the minimum accuracy and the maximum accuracy. As a contrast, our method can achieve much more consistent results. For example, when using the random-selection method on covertype data set, it may result in an accuracy that ranges from 1.63 to 98.37%, while using the proposed method, the accuracy ranges from 95.82 to 96.71%. The average accuracy of our method on covertype data set is 95.91%, which is much higher than the result of 83.05% when using the random-selection method. It is evident that our method can achieve a better and more stable accuracy.

5 Conclusion

To meet the requirements of classifying very large-scale data sets rapidly and accurately at the same time, we proposed a three-stage method. In the first stage, we use a special strategy to run the K-means with a flexible cluster number *k* repeatedly to select representative instances. The selected instances are further reduced by an outlier detection method according to their outlier scores. This approach results in a much smaller representative training instance set for manual labeling. Finally, the labeled instances are used to train a multi-kernel SVM model. The performance of the proposed method is evaluated using different sizes of data sets. The results show that our method can achieve a better accuracy and a faster speed than traditional methods. Our method significantly reduces the human labeling work on very large-scale data sets and achieves a high speed and a good accuracy simultaneously on classification.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (U1509207, 61325019).

Compliance with ethical standards

Conflict of interest Tinglong Tang, Shengyong Chen, Meng Zhao, Wei Huang and Jake Luo declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants performed by any of the authors.

References

Alcalá-Fdez J, Fernández A, Luengo J, Derrac J, García S, Sánchez L, Herrera F (2011) Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. J Multiple-Valued Logic Soft Comput 17:255–287

Arnaiz-González Á, Díez-Pastor J-F, Rodríguez JJ, García-Osorio C (2016) Instance selection of linear complexity for big data. Knowl Based Syst 107:83–95

Bottou L, Lin C-J (2007) Support vector machine solvers. Large Scale Kernel Mach 3(1):301–320

Cavalcanti GDC, Ren TI, Pereira CL (2013) ATISA: adaptive threshold-based instance selection algorithm. Expert Syst Appl 40(17):6894–6900

Chang C-C, Lin C-J (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol (TIST) 2(3):27

Chen H, Zhang Y, Gutman I (2016) A kernel-based clustering method for gene selection with gene expression data. J Biomed Inform 62:12–20

Dean J, Ghemawat S (2008) MapReduce: simplified data processing on large clusters. Commun ACM 51(1):107–113

Dornaika F, Aldine IK (2015) Decremental sparse modeling representative selection for prototype selection. Pattern Recogn 48(11):3714–3727

Hamidzadeh J, Monsefi R, Yazdi HS (2016) Large symmetric margin instance selection algorithm. Int J Mach Learn Cybern 7(1):25–45

Huang Z (1998) Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Min Knowl Discov 2(3):283–304

Khosravani HR, Ruano AE, Ferreira PM (2016) A convex hull-based data selection method for data driven models. Appl Soft Comput 47:515–533

Kim MS (2013) Robust, scalable anomaly detection for large collections of images. In: 2013 International conference on social computing (SocialCom), pp 1054–1058. IEEE

Lichman M (2013) UCI machine learning repository. University of California, School of Information and Computer Science, Irvine, CA. http://archive.ics.uci.edu/ml

Lin W-C, Tsai C-F, Ke S-W, Hung C-W, Eberle W (2015) Learning to detect representative data for large scale instance selection. J Syst Softw 106:1–8

Liu X, Wang L, Yin J, Liu L (2012) Incorporation of radius-info can be simple with SimpleMKL. Neurocomputing 89:30–38

Liu X, Zhou L, Wang L, Zhang J, Yin J, Shen D (2015) An efficient radius-incorporated MKL algorithm for Alzheimer's disease prediction. Pattern Recogn 48(7):2141–2150

Neugebauer J, Kramer O, Sonnenschein M (2016) Improving cascade classifier precision by instance selection and outlier generation. In: ICAART, no. 2, pp 96–104

Olvera-López JA, Carrasco-Ochoa JA, Martínez-Trinidad JF (2010) A new fast prototype selection method based on clustering. Pattern Anal Appl 13(2):131–141

Onan A (2015) A fuzzy-rough nearest neighbor classifier combined with consistency-based subset evaluation and instance selection for automated diagnosis of breast cancer. Expert Syst Appl 42(20):6844–6852

Rakotomamonjy A, Bach FR, Canu S, Grandvalet Y (2008) SimpleMKL. J Mach Learn Res 9(Nov):2491–2521

Rezaei M, Nezamabadi-Pour H (2015) Using gravitational search algorithm in prototype generation for nearest neighbor classification. Neurocomputing 157:256–263

Silva DANS, Souza LC, Motta GHMB (2016) An instance selection method for large datasets based on Markov geometric diffusion. Data Knowl Eng 101:24–41

Stojanović MB, Božić MM, Stanković MM, Stajić ZP (2014) A methodology for training set instance selection using mutual information in time series prediction. Neurocomputing 141:236–245

Sun J, Li H (2011) Dynamic financial distress prediction using instance selection for the disposal of concept drift. Expert Syst Appl 38(3):2566–2576

Triguero I, Derrac JN, GarcíA S, Herrera F (2012) Integrating a differential evolution feature weighting scheme into prototype generation. Neurocomputing 97:332–343

Valero-Mas JJ, Calvo-Zaragoza J, Rico-Juan JR (2016) On the suitability of prototype selection methods for kNN classification with distributed data. Neurocomputing 203:150–160

Whelan M, Le Khac NA, Kechadi M-T (2010) Data reduction in very large spatio-temporal datasets. In: 2010 19th IEEE International workshop on enabling technologies: infrastructures for collaborative enterprises (WETICE). IEEE, pp 104–109

Wilson DR, Martinez TR (2000) Reduction techniques for instance-based learning algorithms. Mach Learn 38(3):257–286

Wu P, Duan F, Guo P (2015) A pre-selecting base kernel method in multiple kernel learning. Neurocomputing 165:46–53

Zhai J, Wang X, Pang X (2016) Voting-based instance selection from large data sets with MapReduce and random weight networks. Inf Sci 367:1066–1077