Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2016) 24: 219 – 233

doi:10.3906/elk-1304-139

Fast and de-noise support vector machine training method based on fuzzy clustering method for large real world datasets

Omid Naghash ALMASI*, Modjtaba ROUHANI
Department of Control and Electrical Engineering, Islamic Azad University, Gonabad, Iran

Received: 15.04.2013 • **Accepted/Published Online:** 30.10.2013 • **Final Version:** 01.01.2016

Abstract: Classifying large and real-world datasets is a challenging problem in machine learning algorithms. Among the machine learning methods, the support vector machine (SVM) is a well-known approach with high generalization ability. Unfortunately, while the number of training data increases and the data contain noise, the performance of SVM significantly decreases. In this paper, a fast and de-noise two-stage method for training SVMs to deal with large, real-world datasets is proposed. In the first stage, data that contain noises or are suspected to be noisy are identified and eliminated from the genuine training dataset. The process of elimination and identification is based on the movement of the center of the convex hull data in the training dataset. The convex hull data are computed via the QHull algorithm. On the other hand, the well-known fuzzy clustering method (FCM) is applied to compress and reduce the size of the training dataset. Finally, the reduced and purified cluster centers are used for training the SVM. A set of experiments is conducted on the four benchmarking datasets of the UCI database. Moreover, the amount of training time and the generalization of the proposed approach are compared with FCM-SVM and normal SVM. The results indicate that the proposed method reduces the amount of training time and has a considerable success in removing noisy data from the training dataset. Therefore, the proposed method can achieve a higher generalization performance in comparison with the other methods in large, real-world datasets.

Key words: Support vector machine, fuzzy clustering method, convex hull, QHull algorithm, reduction set method, noisy training dataset

1. Introduction

In recent years, the widespread use of computers and the development of technology have led to remarkable progress in the production and storage of numerical data. Consequently, large datasets in various fields are produced and can be found in commercial exchange, agricultural trade, the internet, traffic, telecommunications, astronomy, and medical services, to name a few. For this reason, there is an essential requirement for developing fast and accurate learning machine algorithms and data mining approaches in the classification of large, real-world datasets.

The presence of noise is highly likely in real-world large datasets. Noise will confuse a machine learning algorithm in the training phase. Accordingly, the accurate performance and generalization ability are noticeably reduced [1–3]. Thus, an important phase associated with the use of machine learning algorithms is removing the noise from the training dataset [1,4]. A key factor in removing noisy data from the original training dataset in the classification problem is how to detect and remove noisy data from pure data [1,5].

*Correspondence: o.almasi@ieee.org

Due to the high generalization performance and great mathematical background, the support vector machine (SVM) is one of the well-known machine learning algorithms in the classification problem [6,7]. The present mathematical formulation of SVM was obtained through a three-stage evolution. The first phase was started in 1963 with introducing the idea of Vapnik and Lerner to construct the optimal hyperplane, i.e. a linear classifier with the largest margin separating the linear separable training data. Then, by a three-member team effort consisting of Guyon, Boser, and Vapnik, the construction idea of an optimal hyperplane was extended to the feature space by using a kernel function. Finally, by Cortes and Vapnik, the soft margin formulation of SVM for noisy data was introduced and the final formulation was obtained. This form of SVM formulation is suitable and general for all real-world datasets, linear and nonlinear, and separable and nonseparable [6–8].

The SVM is highly sensitive to a number of training data and to the presence of noises in the training dataset [9-13]. Training SVM is equivalent to solving a convex quadratic problem, with the significant computational benefit of not getting stuck in local minima. But when the size of the training dataset is increased, the amount of training time goes up, and much worse, the QP kernel matrix size is enlarged and cannot be stored in memory. Hence, training a SVM is a slow process and turns into a serious challenging problem for large datasets on a large scale. The presence of noisy data and outliers is inevitable in large and real-world datasets because it is impossible to avoid hardware failure of measuring tools such as sensors, transmitters, and transducers, or calibration errors of measurement devices and programming errors, all of which lead to the creation of noisy and outlier data. In the construction of an optimal hyperplane in two-class classification problems, the presence of noisy data in training datasets is one of the main reasons for reducing the accuracy performance [5,9–11,13,14].

In order to improve the accuracy of SVM and to reduce SVM training time in noisy and large datasets, many researchers have tried two different isolation categories: eliminating noisy data from the training dataset and reducing the size of the training dataset.

A. Methods for reducing SVM training time

1) Algebraic methods: A group of researchers tried to split the big SVM QP problem into smaller size QP problems and then combined the final solutions of these QPs to obtain the optimal hyperplane solution [15,16]. As an example of this type of methods, [17] proposed a fast and efficient method named sequential minimal optimization (SMO). The other efforts of researchers in the field of algebraic methods are available in [18], [19], and [20]; they optimized and modified the SMO with regard to updating two violation parameters of KKT conditions simultaneously, presented a new stop condition for improving the convergence speed of the SMO algorithm, and proposed a multiprocessor parallel algorithm to accelerate the standard SMO algorithm, respectively.
2) Geometrical methods: Some researchers restricted the upper bound of the convex hull in the feature space and constructed a linear hyperplane separator [2,21]. The other types of geometric approaches are found in [22] and [23], where the data points lying on the convex hull of the entire dataset in each of the classes were first computed and then used as a training dataset for training SVMs and FSVMs.
3) Hybrid methods in reducing SVM training time: Another group of researchers employ clustering methods, i.e. hierarchical clustering [24], minimum enclosing ball clustering [25], and fuzzy $k-$means clustering [26–31], to make an effort to extract a training dataset with smaller size than the original training dataset. They believe these data are the most informative and highly representative in a large dataset for finding support vectors. Thus, by using the selected training dataset, the training time is decreased while the performance remains high.

B. Methods for reducing the effect of noises

1) Statistical methods: Statistics is a traditional tool for noise detection. [32] proposed an iterative algorithm named C4.5 to eliminate noisy data from the other data. John et al. used an information criterion to measure the similarity of samples, and then a decision was made by a human expert on whether the data contained noise and needed to be eliminated or not [33]. The main drawback is that some distribution of data is needed to assume these methods.

2) Fuzzy SVMs methods: The researchers in [14] and [34] reformulate the SVM and make the fuzzy SVM (FSVM). The main idea in FSVM for reducing the effect of noise is to apply a fuzzy membership (importance weight) to each sample in the training dataset such that different samples can make different contributions in the construction of the SVM hyperplane. A key point in these methods is how to determine weights for a dataset. Examples of these methods are given in [12–14], where smaller weight or even zero weight was assigned to noisy and outlier data to reduce noisy data or remove them.

3) Hybrids methods: A group of researchers proposed multiple-stage hybrid methods, which combine a clustering approach or a filter with SVM. Brodley et al. proposed a filter to clean noisy data and then used the pure data as a training dataset for training a classifier [35]. In [36], a KFCM-clustering-based FSVM algorithm (KFCM-FSVM) was proposed to deal with the classification problems with outliers or noises.

In this paper, a fast and insensitive to noise training method for dealing with large and real-world datasets is proposed. The method has two stages. The first stage removes noisy and outlier data based on the displacement of the center of the convex hull data in the training dataset. In the second stage, the FCM algorithm reduces the size of pure data obtained from the first stage. Finally, the purified cluster centers are used for training SVM. Therefore, the proposed method simultaneously decreased the training time of SVM and enhanced the accuracy of the separating hyperplane in SVM.

The paper is organized as follows. Section 2.1 briefly recalls existing SVM mathematical formulation and Sections 2.2 and 2.3 describe FCM and the convex hull, respectively. Section 3 presents the proposed hybrid model. In Section 4, the proposed approach faces a serious challenge based on the well-known large and real-world datasets of the UCI database and discusses the results in details. Finally, Section 5 draws relevant conclusions.

2. Basic background

With the development of software and hardware, large volumes of large datasets are generated. Classification is a process for analyzing data and extracting a model from data in order to describe the concept of data and estimate the future behavior of them (their corresponding class labels). The discrete models are obtained from the classification methods where the output is either 1 or –1. Predicting the future behavior of a test dataset by a classifier is equivalent to assigning a class label to each of the data points with an unknown class label in the test dataset.

In recent years, SVM has gained a lot of attention in comparison with other machine learning algorithms, and has been used in many real-world applications. Unfortunately, due to the large QP optimization of SVM, the training phase is slow and the memory requirement of storing the kernel matrix is large. This is a serious challenge in the real-world application of SVMs.

In addition, it is noteworthy that large datasets obtained from real-world applications contain noisy data. Noisy data will complicate the training phase of SVM, and hence a substantial reduction in the modeling performance and the generalization ability of SVM are observed. Generally, it can be concluded that a fast and robust method for analyzing and building an accurate model from large and noisy datasets is absolutely essential.

In Section 2.1 SVM, as one of the well-known machine learning methods, is briefly reviewed. Section 2.2 describes FCM as a kind of soft clustering and also a nice reduction method. Finally, in Section 2.3 convex hull and its algorithm is defined.

2.1. Support vector machine

The necessary formulation of SVM for classification problems is reviewed in this section. Assume a set of two classes of labeled training points $(x_i y_i)$ is given. For $i = 1, \ldots, n$, each training point $x_i \in R^n$ belongs to one of the two classes in accordance with label $y_i \in \{-1, 1\}$. The optimal hyperplane is obtained by solving a quadratic optimization problem in Eq. (1) (known as primal form), whose number of variables is as large as the training data size n.

$$\min \varphi(w, \xi) = \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \tag{1}$$

s.t.

$$\begin{aligned} & y_i \left(w^T . \, x_i + b \right) \geq 1 - \xi_i, \quad i = 1, 2, \ldots, n \\ & \xi_i \geq 0, \qquad\qquad\qquad\quad i = 1, 2, \ldots, n \end{aligned} \tag{2}$$

where ξ_is are slack variables that represent a violation of the pattern separation condition. The user defined parameter C is regarded as a regularization parameter controlling the model complexity. For nonlinear separable data, the kernel trick is utilized to map the input space into a high dimensional space named feature space. The optimal hyperplane is obtained in the feature space. The primal optimal problem of Eq. (1) can be transformed into a dual form as:

$$\max Q(\alpha) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, x_j) - \sum_{j=1}^{n} \alpha_j$$

s.t.

$$\sum_{j=1}^{l} \alpha_i y_i = 0 \tag{3}$$

$$0 \leq \alpha_i \leq C, \qquad i = 1, \ldots, n$$

where $k(.,.)$ is a kernel function. In the practical applications of SVMs, there are several frequently used substitutions for selecting the kernel function $k(.,.)$. Some of the conventional kernel functions are listed in Table 1. In this table, σ and d are constants and those parameters must be set by a user. For MLP kernel a suitable choice for β_0 and β_1 is needed to enable the kernel function satisfying the Mercer condition [6,8].

Those kernel parameters highly affect the generalization performance and also the model complexity of the resulting SV machine. Kernel parameters are implicitly characterizing the geometric structure of data in the feature space. In the feature space the data become linearly separable, where the maximal margin of separation between the two classes is reached. The selection of kernel parameters will change the shape of the separating

Table 1. The conventional kernel functions.

Name	Kernel function expression
Linear kernel	$k(x_i, x_j) = x_i^T x_j$
Polynomial Kernel	$k(x_i, x_j) = (t + x_i^T x_j)^d$
RBF kernel	$k(x_i, x_j) = exp(-\|x_i - x_j\|^2 / \sigma^2)$
MLP kernel	$k(x_i, x_j) = tanh(\beta_0 x_i^T x_j + \beta_1)$

surface in the input space. The optimal choice of a regularization parameter and kernel parameters is called the SVM model selection problem [1,7,37]. In [38] and [39], many approaches were suggested for the optimal model selection problem. Furthermore, in Eq. (2) $\alpha = (\alpha_1, \ldots, \alpha_n)$ is the vector of nonnegative Lagrange multipliers. The solution vector $\alpha = (\alpha_1, \ldots, \alpha_n)$ is sparse, i.e. $\alpha_i = 0$ for most indices of the training dataset. This is the so-called SVM sparseness property. The points x_i corresponding to nonzero α_i are called support vectors. Therefore, the points x_i corresponding to $a_i = 0$ have no participation in the construction of the optimal hyperplane and only a part of the training dataset, i.e. the support vectors, constructs the optimal hyperplane. Let v be the index set of support vectors, then the optimal hyperplane is:

$$f(x) = \sum_{i \in \nu}^{\#sv} \alpha_i y_i k(x_i, x_j) + b = 0 \tag{4}$$

and the resulting classifier is:

$$y(x) = sgn\left[\sum_{i \in \nu}^{\#sv} \alpha_i y_i k(x_i, x_j) + b\right] \tag{5}$$

where b is easily determined by KKT conditions. In regards to Eq. (3), in the SVM training phase, a big kernel matrix is required, whose rows are equivalent to the number of training data. Thus, in real application, the storage of this kernel matrix has a large computational cost. Time complexity and space storage for the SVM training are $o(l^3)$ and $o(l^2)$, respectively [9,10].

2.2. Fuzzy clustering method

Bezdek introduced fuzzy clustering by using an objective function in 1981 [29]. The FCM is clustering data such that data within each cluster are the most similar to each other and data in different clusters are as dissimilar as possible [29,40]. The FCM is a soft clustering method; thus, each item/piece of data has a specific degree of freedom that allows it to belong to all the clusters rather than only one cluster. The fuzzy clustering algorithm is defined as the minimization of the following objective function:

$$Min(J, U) = \sum_{k=1}^{l} \sum_{i=1}^{C} u_{ik}^m d(x_i, v_k)$$

s.t.

$$\sum_{i=1}^{c} u_{ij} = 1\ ,\ 0 \leq u_{ij} \leq 1 \tag{6}$$

where l and c are the number of training datasets and clusters, respectively. The proposed method is assumed to create scattered clusters, so data being close to a distinctive hyperplane have a higher chance of being selected

as a cluster center. In Eq. (6), m determines the data dispersion degree in all the clusters. With an equal number of clusters and an increasing fuzziness parameter m, the value of the objective function is reduced and the centers of clusters scatter less, and vice versa. Therefore, to increase the chance of misclassified data being the centers of clusters, all the clusters should be as scattered as possible. This is only possible with a proper choice of parameter m. In this study, m is equal to 1.2. In Eq. (6), $d(x_j v_i)$ is the measure of similarity between jth data and the ith cluster, which is defined in the Euclidian norm as follows:

$$d\left(x_i, v_k\right) = \|x_i - v_k\| = \left[\sum_{j=1}^{d} \left(x_{kj} - v_{ij}\right)^2\right]^{1/2} \tag{7}$$

where v_k and u_{ij} are the cluster center and the membership degrees of data to each cluster, updated based on Eq. (8) and Eq. (9), respectively.

$$v_{ij} = \frac{\sum_{k=1}^{l} u_{ik}^m x_{ki}}{\sum_{k=1}^{l} u_{ik}^m} \tag{8}$$

where $= 1, \ldots$, and M is the number of features of data.

$$u_{ik} = \left[\sum_{j=1}^{c} \left(\frac{d_{ik}}{d_{jk}}\right)^{2/(M-1)}\right]^{-1} \tag{9}$$

The flowchart of the FCM is shown in Figure 1.

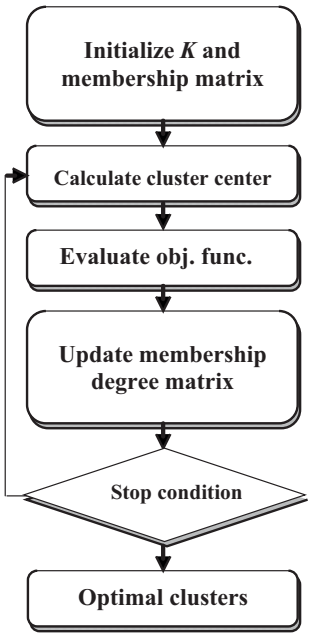

Figure 1. The FCM flowchart.

2.3. Convex hull

The convex hull of a training dataset X in R^n is the smallest convex set that contains the entire training dataset [2,41]. The convex hull of training points in each class shows that finding the nearest points of two convex hulls is equivalent to finding the separating hyperplane with the maximum margin in the SVM problem [42]. An algebraic view of the convex hull for X is the set of all the combinations of samples in X. It is formulated as follows:

$$Conv\,(X) = \left\{ \sum_{i=1}^{l} \alpha_i x_i \,|\, x_i \in X, \alpha_i \geq 0,\ \sum_{i=1}^{l} \alpha_i = 1\,,\ i = 1, \ldots, l \right\} \tag{10}$$

where l is the total number of finite training samples x_i in X and α_i are nonnegative parameters that must satisfy $\sum_{i=1}^{l} \alpha_i = 1$ condition. Many algorithms have been suggested for computing the convex hull [42,43]. Among them, the QHull algorithm has a lesser computational complexity encounter in large datasets in R^n [44,45]. It has the expected time complexity $o(nlogn)$ [45].

3. Fast and de-noise SVM training method

The separation hyperplane of the SVM is constructed by only a part of the training dataset, the support vectors, which are obtained from the training phase. One effective method to speed up the SVM training phase is to reduce the input data size of the training phase. Another one is to select a part of the training data as a representative of the entire training dataset. The proposed method uses the FCM for selecting a part of the training dataset (data reduction stage), while holding meaningful data in the training dataset. Then, the obtained cluster centers are used as a reduced dataset for training the SVM. The new training dataset boosts up the SVM training phase, but decreases the accuracy of the SVM because the cluster centers are calculated by averaging over the whole data of each cluster in Eq. (8). Thus, it is obvious that FCM is highly sensitive to noisy and outlier data points. The noisy data samples contribute extremely effectively to the construction of cluster centers, so using cluster centers without employing a de-noise process of these data is useless and makes the reduction method ineffective. The reduced dataset is not an accurate representation of the entire training dataset in large and real-world applications. Therefore, it is necessary to eliminate the noisy data and the suspected noisy data from the other data in the training phase. Generally, noisy data in classification problems could be organized in three groups [4,35,46,47]: 1) data whose corresponding labels include noise (labeling error); 2) data whose attributes are noisy; and 3) data that have both the first and the second group of noises, i.e. noise in class labels and their attributes.

Noisy data belong to either of the mentioned noise groups and have a lesser similarity to their own data with a corresponding class label of +1 or –1. These data commonly lie on the boundary between the convex hulls of two classes. This is a key point, and based on it, the proposed method is able to remove the suspected noisy data from the training dataset.

The proposed method has two stages. In the first stage, an iterative algorithm is used for removing noisy data from the training dataset. In the second stage, the FCM is employed as a reduction method to reduce the number of training data.

The iteration algorithm for the positive class is as follows: in each iteration, the convex hull data is computed by the QHull algorithm. Then, in regards to the key point, the boundary data suspected to contain noise are eliminated. Then, for the remaining training dataset, the class center is calculated and the termination

condition in Eq. (11) is examined. If it is not satisfied, the iteration is repeated. The iteration algorithm for the negative class is the same as the one for the positive class.

The stop criterion is based on the displacement of the class center position of each class. The stop condition of the iterative algorithm, by selecting an appropriate parameter named δ, is defined as follows:

$$\|ClassCenter\,(i) - ClassCenter(i-1)\| \leq \delta_x \tag{11}$$

where i is the index of each iteration and δ_x is equal to one of the values δ_P and δ_N. The parameters δ_P and δ_N are used for restricting the displacement of the positive class label, i.e. $y = +1$, and the negative class label, i.e. $y = -1$, respectively.

In order to control the removal rate of boundary data containing noise or suspected noise, two ranges are defined for the class centers, i.e. $\delta_x = [\delta_p\, \delta_N]$. By tuning these ranges, the movements of the class center positions are restricted. To illustrate the concept of the class center of the convex hull data, a synthetic dataset containing noise is generated in Figure 2. It will be used for further explanation of the first stage of the proposed method and its significance to determine a proper value for δ_x and to clarify how the termination condition is worked out.

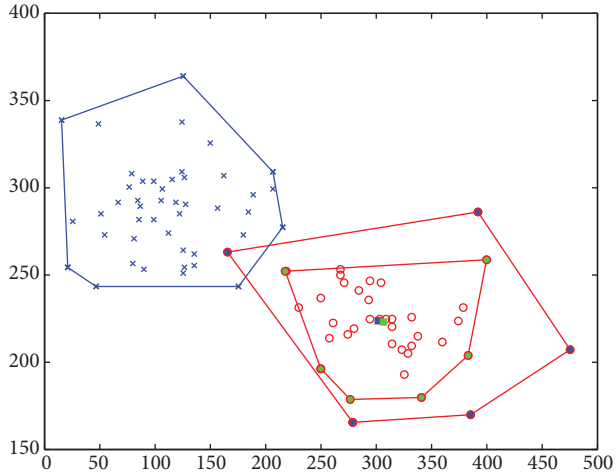

Figure 2. The effectiveness of eliminating noisy data in the class center of the red synthetic dataset.

The red dataset is marked with hollow circles (O) and the blue class is specified with crosses ($\times$) in Figure 2. Some noisy data from the three noise groups exist in the red dataset. The red circles, which are bolded with purple color, are made of the primal convex hull data of the red dataset. The position of the class center corresponding to the primal convex hull data, that is the red circle filled with purple color, is shown with a purple square. After running the first stage of the proposed method, data that make up the convex hull are removed. It is obvious that the convex hull data are more likely to be noisy data or have data that probably contain much noise.

After eliminating the convex hull data of the primal convex hull, another convex hull is computed for existing data and their convex hull data is determined. Then, similar to the previous iteration, the class center is calculated for the obtained convex hull data. In Figure 2 the second convex hull data are shown with red circles filled by a green color and their corresponding class center colored with a green square.

By comparing the purple and the green center positions with each other, the effectiveness of removing noisy data in the shift of the center position is demonstrated. Eliminating noisy data caused the movement of

the class center position from the purple to the green. In fact, the key parameter δ_x represents the differences arising from the elimination of noisy data in the obtained class centers. Thus, the selection of $\delta_x = [\delta_p\, \delta_N]$ has a major effect on the position of each class center. If it is too high, the first stage of the proposed algorithm, in addition to removing noisy data, will eliminate meaningful data (without noise) too; and if it is too low for parameter $\delta_x = [\delta_p\, \delta_N]$, the efficacy of the first stage of the proposed method will be discarded.

The proposed algorithm has two separate parameters, δ_p and δ_N, in δ_x to adjust the elimination rate of noisy data for each class of datasets.

The two separate parameters of δ_x produced a considerable ability for the expert user to use the proposed method in analyzing imbalance datasets that need different values of $\delta_x = [\delta_p\, \delta_N]$, or datasets with different amounts of noise in their classes.

In the second stage of the proposed method, the FCM is used to reduce the number of purified data in each class of datasets. In other words, the FCM chooses informative representative data for each class of datasets. After applying the FCM, the purified cluster centers are calculated based on Eq. (8) and are used as a reduced training dataset for training the SVM. The flowchart of the proposed method is presented in Figure 3.

4. Experiments

4.1. Experimental conditions

To evaluate the validity of the proposed method, large real-world datasets of the UCI database were used. The specifications of the datasets are shown in Table 2.

Table 2. Dataset descriptions.

Classes		Attributes	Records	Datasets name	Datasets indices
Pos.	Neg.				
3		4	625	Balance	1
1	2,3				
2		2	862	Four class	2
1	–1				
2		4	4000	SVMguide1	3
1	–1				
2		2	10,000	Banana	4
1	–1				

A comparative study was carried out between the normal SVM [6], which does not use a noisy data removal process and a reduction method for the training data size, and the FCM-SVM based on [26,27], which only uses a reduction method for reducing the number of training data. The cluster centers are used as a training dataset for training the SVM in the proposed method. As mentioned in Section 2.2, there is not a specific method to select the appropriate number of clusters. In this study, similar to other studies, the FCM algorithm used the method of trial and error to conjecture the optimal number of clusters [5,26–29].

For implementing the comparative study, a PC and MATLAB (R2008b) software was employed. Table 3 shows the hardware configuration of the PC. The Gaussian kernel with the same value for its parameter is used in all experiments. The regularization parameter of SVM according to different datasets is set equal in all the approaches. All the parameters used in the proposed method are presented in Table 4.

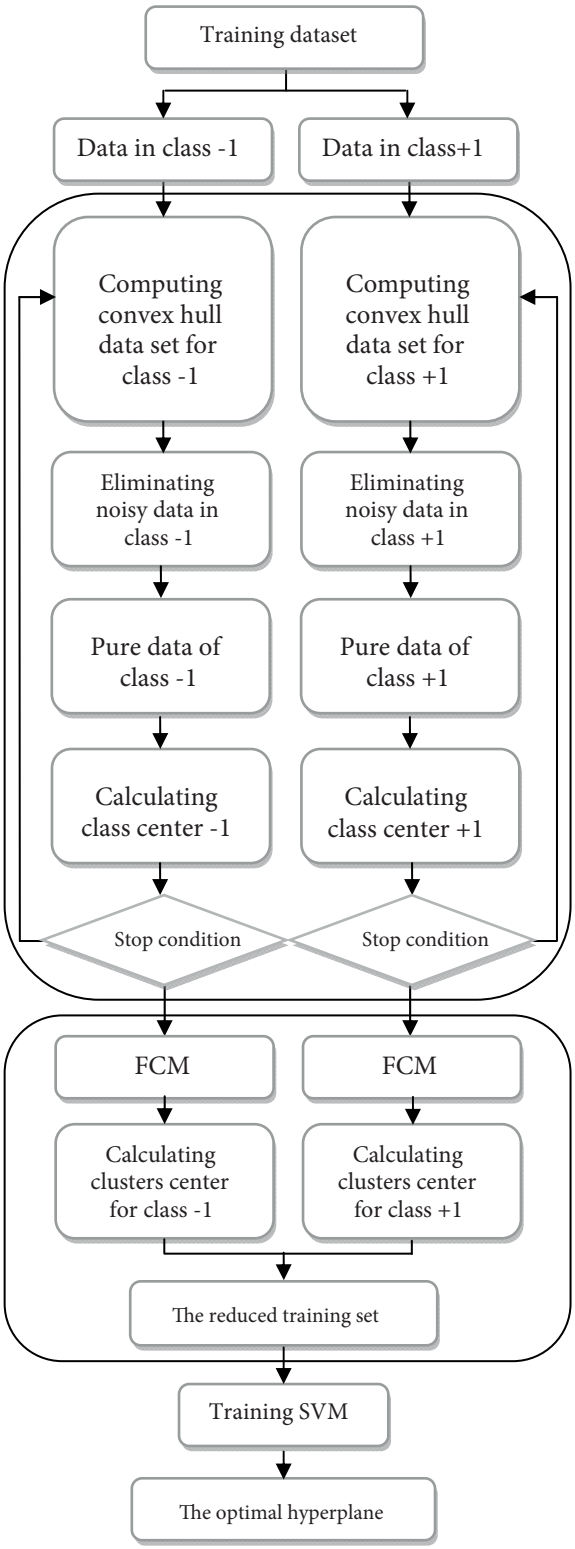

Figure 3. Flowchart of the proposed method.

Table 3. Hardware configuration.

64 AMDTurion	CPU
2 GB	RAM
32 bit Vista	OS

Table 4. The parameters of the proposed method.

4	3	2	1	Datasets indices
80	220	59	50	# Clusters for each class
0.01	1	0.3	0.01	$\delta_{Positive}$
0.01	1.5	0.5	0.001	$\delta_{Negative}$
1	1	1	1	Kernel parameter
1000	10,000	10,000	10,000	Regulation parameter

The appropriate choice of $\delta = [\delta_p\, \delta_N]$ is done by the user to perform multiple tests (trial and error testing) and a rudimentary knowledge about the amount of noise in each class of dataset is obtained. The suitable set of parameter δ is crucial in forming the separating hyperplane. For this reason, choosing a large value for the parameter $\delta = [\delta_p\, \delta_N]$ causes the incorrect elimination of meaningful and informative data of each class. By contrast, selecting a small value for the parameter $\delta = [\delta_p\, \delta_N]$ affects the first stage of the proposed method for removing noisy and suspected noisy data from the training dataset.

4.2. Experimental results and discussion

The generalization performance and the amount of training time in all the experiments are given in Tables 5 and 6, respectively.

Table 5. A comparison of accuracies.

Accuracy (%)				
Datasets indices	1	2	3	4
Normal SVM [6]	95.19	99.48	95.35	94.58
FCM-SVM [26,27]	96.15	98.78	94.69	95.11
The proposed method	99.22	99.76	95.80	95.28

Table 6. A comparison of time.

Time training (s)				
Datasets indices	1	2	3	4
Normal SVM [6]	1.79	2.97	113.02	76.40
FCM-SVM [26,27]	3.34	5.48	79.30	40.73
The proposed method	4.08	5.70	62.47	27.13

Balance dataset: According to Table 5, the proposed technique eliminates noisy and suspected noisy data effectively. So, the generalization performance of the proposed method is higher than that of the normal SVM and FCM-SVM methods. But Table 6 shows that the proposed method has more training time in comparison with the two other methods.

Fourclass dataset: The proposed algorithm is successful in correctly removing the noisy and suspected noisy data; however, the amount of training time is a little more in comparison with that of the other two mentioned approaches.

SVMguide1 dataset: This dataset has a size of 6.4 and 4.64 times the size of the two other datasets, i.e. Balance and Fourclass, respectively. Tables 5 and 6 show that the proposed method not only improved the generalization performance, but also reduced the amount of training time in comparison with normal SVM and FCM-SVM.

Banana dataset: This is the largest dataset of the experiment. The size of the Banana dataset is 16, 13.12, and 2.5 times the size of Balance, Fourclass, and SVMguide1, respectively. Tables 5 and 6 disclose some interesting facts that the proposed method is able to simultaneously obtain a robust classifier with higher generalization performance and less training time in the training phase of SVM.

Generally, it can be concluded from the results of this research that the proposed method has more accurate generalization performance in all experiments.

It consumes more time in its training phase on the small size datasets; however, when the size of the training dataset is increased, the proposed method needs a lesser amount of training time with large and real-world datasets when compared with the two other methods. Figures 4 and 5 show the results of the comparative study for the proposed method and the two other methods based on the literature [6,26,27].

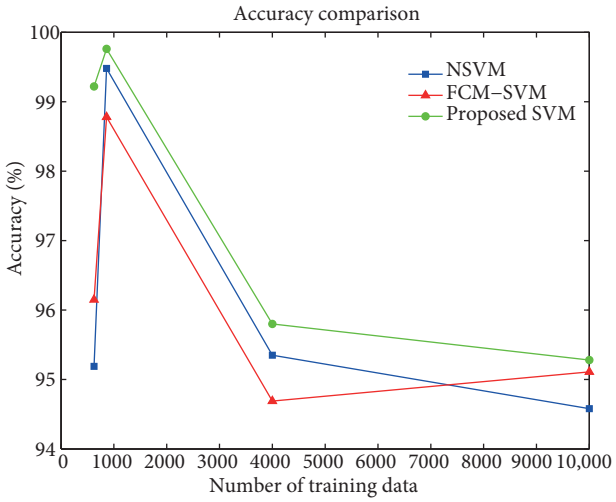

Figure 4. A comparison of accuracies. Normal SVM (blue), FCM-SVM (red), and the proposed method (green).

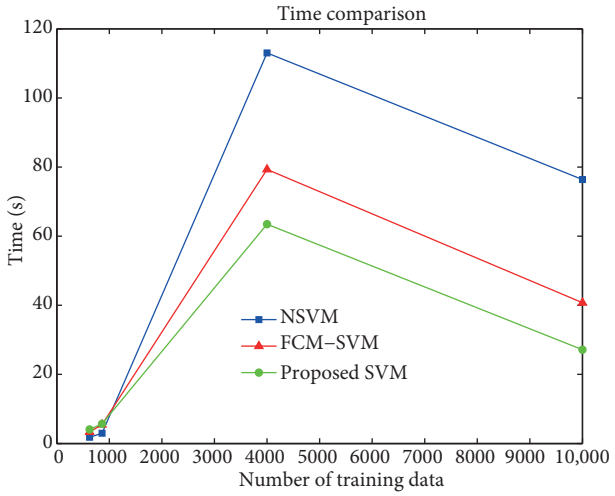

Figure 5. A comparison of time. Normal SVM (blue), FCM-SVM (red), and the proposed method (green).

5. Conclusion

The SVM is a machine learning method with a powerful mathematical foundation. Although SVM demonstrates good generalization performance, when the number of training datasets is increased or the datasets contain noise, some potential difficulties arise. Accordingly, the SVM accuracy drops and the training time increases, making SVM a slow machine learner.

In this paper, a fast and de-noise SVM training method for training SVM in large and real-world datasets is proposed. This method is based on the convex hull data, the displacement of each class center, and FCM.

To illustrate the effectiveness of the proposed approach, some experiments were performed on large datasets of the UCI database. Moreover, a comparative study was carried out to show the superior performance of proposed method to normal SVM and FCM-SVM.

The results indicate that the proposed method, by extracting informative and meaningful data, can

enhance the training time of SVM in large and real-world datasets. Furthermore, the proposed method has an effective performance in removing noisy and suspected noisy data. Therefore, the SVM achieves higher generalization performance and makes an accurate and robust separating hyperplane in large and noisy datasets.

References

[1] Hulse JV, Khoshgoftaar TM. Knowledge discovery from imbalanced and noisy data. Data Knowl Eng 2009; 68: 1513–1542.

[2] Mavroforakis ME, Theodoridis S. A geometric approach to Support Vector Machine (SVM) classification. IEEE T Neural Networ 2006; 17: 671–682.

[3] Angelova A, Abu-Mostafa Y, Perona P. Pruning training sets for learning of object categories. Proc Cvpr IEEE 2005; 494–501.

[4] Zhu X, Wu X. Class noise vs. attribute noise: a quantitative study of their impacts. Artif Intell Rev 2004; 22: 177–210.

[5] Yang X, Zhang G, Lu J, Ma J. A kernel fuzzy c-means clustering-based fuzzy support vector machine algorithm for classification problems with outliers or noises. IEEE T Fuzzy Syst 2011; 19: 105–115.

[6] Vapnik V. Statistical Learning Theory. New York, NY, USA: Wiley, 1998.

[7] Cristianini N, Taylor JS. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge, UK: Cambridge University Press, 2000.

[8] Bordes A, Ertekin S, Weston J, Bottou L. Fast kernel classifiers with online and active learning. J Mach Learn Res 2005; 6: 1579–1619.

[9] Angiulli F, Astorino A. Scaling up support vector machines using nearest neighbor condensation. IEEE T Neural Networ 2010; 21: 351–357.

[10] Dong JX, Krzyzak A, Suen CY. Fast SVM training algorithm with decomposition on very large data sets. IEEE T Pattern Anal 2005; 27: 603–618.

[11] Xinjun P. A nu-twin support vector machine (nu-TSVM) classifier and its geometric algorithms. Inform Sciences 2010; 180: 3863–3875.

[12] Tang WM. SVM with a new fuzzy membership function to solve the two-class problems. Neural Process Lett 2011; 34: 209–219.

[13] Lu YL, Li L, Zhou MM, Tian GL. A new fuzzy support vector machine based on mixed kernel function. In: IEEE International Conference on Machine Learning and Cybernetics; 12–15 July 2009; Baoding, China: IEEE. pp. 12–15.

[14] Lin CF, Wang S. Fuzzy support vector machines. IEEE T Neural Networ 2002; 13: 464–471.

[15] Osuna E, Freund R, Girosi F. An improved training algorithm for support vector machines. Proceedings of Neural Networks for Signal Processing 1997; 276–285.

[16] Vapnik V. Estimation of Dependences Based on Empirical Data. Berlin, Germany: Springer-Verlag, 1982.

[17] Platt J. Fast training of support vector machines using sequential minimal optimization. In: Schölkopf B, Burges C, Smola A, editors. Advances in Kernel Methods-Support Vector Learning, Cambridge, MA, USA: MIT Press, 1999. pp. 185–208.

[18] Keerthi SS, Shevade SK, Bhattachayya C, Murth KRK. Improvements to Platt's SMO algorithm for SVM classifier design. Neural Comput 2001; 13: 637–649.

[19] Zhiyong D, Zuolin D, Peixin Q, Xianfang W. Fuzzy support vector machine based on improved sequential minimal optimization algorithm. In: IEEE International Conference on Computer and Communication Technologies in Agriculture Engineering; 12–13 June 2010; Chengdu, China: IEEE. pp. 152–155.

[20] Peng P, Ma QL, Hong LM. The research of the parallel SMO algorithm for solving SVM. In: IEEE International Conference on Machine Learning and Cybernetics; 12–15 July 2009; Baoding, China: IEEE. pp. 1271–1274.

[21] Liu Z, Liu JG, Pan C, Wang G. A novel geometric approach to binary classification based on scaled convex hulls. IEEE T Neural Networ 2009; 20: 1215–1220.

[22] Hong Z, Xiao W, Long XH, Lei LY, Wen Q. Fast SVM training based on thick convex-hull. In: IEEE Congress on Image and Signal Processing; 27–30 May 2008; Sanya, China: IEEE. pp. 584–587.

[23] Liu H, Xiong S, Chen Q. Fuzzy support vector machines based on convex hulls. In: IEEE International Symposium on Knowledge Acquisition and Modeling; 21–22 December 2008; Wuhan, China: IEEE. pp. 920–923.

[24] Xu R, Wunsch D. Survey of clustering algorithms. IEEE T Neural Networ 2005; 16: 645–678.

[25] Cervantes J, Li X, Yu W, Li K. Support vector machine classification for large data sets via minimum enclosing ball clustering. Neurocomputing 2008; 71: 611–619.

[26] Cervantes J, Li X, Yu W. Support vector machine classification based on fuzzy clustering for large data sets. In: MICAI 2006: Advances in Artificial Intelligence. Berlin, Germany: Springer-Verlag, 2006. pp. 572–582.

[27] Li X, Cervantes J, Yu W. A novel SVM classification method for large data sets. In: IEEE International Conference on Granular Computing; 14–16 August 2010; Silicon Valley, CA, USA: IEEE. pp. 297–302.

[28] Saha I, Ujjwal M, Sanghamitra B, Dariusz P. Improvement of new automatic differential fuzzy clustering using SVM classifier for microarray analysis. Expert Syst Appl 2011; 38: 15122–15133.

[29] Bezdek JC. Pattern Recognition with Fuzzy Objective Function Algorithms. New York, NY, USA: Plenum Press, 1981.

[30] Carvalho D, De AT F, Lechevallier Y, De Melo FM. Relational partitioning fuzzy clustering algorithms based on multiple dissimilarity matrices. Fuzzy Set Syst 2013: 215: 1–28.

[31] Zhou SM, Gan JQ. Constructing L2-SVM-based fuzzy classifiers in high-dimensional space with automatic model selection and fuzzy rule ranking. IEEE T Fuzzy Syst 2007; 15: 398–409.

[32] Zhu X, Wu X, Yang Y. Error detection and impact-sensitive instance ranking in noisy data. In: AAAI National Conference on Artificial Intelligence; 25–29 July 2004; San Jose, CA, USA: AAAI. pp. 378–384.

[33] John GH. Robust decision trees: removing outliers from databases. Lect Notes Artif Int 1995; 174–179.

[34] Inoue T, Abe S. Fuzzy support vector machines for pattern classification. In: The International Joint Conference on Neural Networks; 15–19 July 2001; Washington DC, USA: IEEE. pp. 1449–1454.

[35] Brodley CE, Friedl MA. Identifying mislabeled training data. J Artif Intell Res 1999; 11: 131–167.

[36] Yang X, Zhang G, Lu J, Ma J. A kernel fuzzy c-means clustering-based fuzzy support vector machine algorithm for classification problems with outliers or noises. IEEE T Fuzzy Syst 2011; 19: 105–115.

[37] Khoshgoftaar TM, Seliya N. The necessity of assuring quality in software measurement data. In: Proceedings of the 10th International Software Metrics Symposium; 14–16 September 2004; Chicago, IL, USA: IEEE. pp. 119–130.

[38] Sun J, Zheng C, Li X, Zhou Y. Analysis of the distance between two classes for tuning SVM hyper-parameters. IEEE T Neural Networ 2010; 21: 305–318.

[39] Peng X, Wang Y. A geometric method for model selection in support vector machine. Expert Syst Appl 2009; 36: 5745–5749.

[40] Pal NR, Bezdek JC. On cluster validity for the fuzzy c-means model. IEEE T Fuzzy Syst 1995; 3: 370–379.

[41] Goodrich B, Albrecht D, Tischer P. Algorithms for the Computation of Reduced Convex Hulls. Berlin, Germany: Springer-Verlag, 2009.

[42] Bennett KP, Bredensteiner EJ. Duality and geometry in SVM classifiers. In: Proceedings of the 17th International Conference on Machine Learning; 2000; San Francisco, CA, USA. pp. 57–64.

[43] Preparata FP, Hong SJ. Convex hulls of finite sets of points in two and three dimensions. Commun ACM 1977; 20: 87–93.

[44] Zhou X, Wenhan J, Yingjie T, Yong S. Kernel subclass convex hull sample selection method for SVM on face recognition. Neurocomputing 2010; 73: 2234–2246.

[45] Barber CB, Huhdanpaa H. The quickhull algorithm for convex hulls. ACM T Math Software 1966; 22: 469–483.

[46] Hulse JV, Khoshgoftaar TM, Huang H. The pairwise attribute noise detection algorithm. Knowl Inf Syst 2007; 11: 171–190.

[47] Khoshgoftaar TM, Zhong S, Joshi V. Enhancing software quality estimation using ensemble-classifier based noise filtering. Intell Data Anal 2005; 9: 3–27.