SVM-Based Incremental Learning Algorithm for Large-Scale Data Stream in Cloud Computing

Ning Wang[1,2], Yang Yang[1], Liyuan Feng[1], Zhenqiang Mi[1], Kun Meng[3], Qing Ji[1]
[1] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, 100083, China [e-mail: wangning200658@126.com, mizq@ustb.edu.cn]
[2] Department of Mechanical and Electrical Engineering, Yantai Engineering & Technology College, Yantai, 264006, China
[3] Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China
*Corresponding author: Zhenqiang Mi

Received April 9, 2014; revised June 29, 2014; accepted August 22, 2014; published October 31, 2014

Abstract

We have witnessed the rapid development of information technology in recent years. One of the key phenomena is the fast, near-exponential increase of data. Consequently, most of the traditional data classification methods fail to meet the dynamic and real-time demands of today's data processing and analyzing needs--especially for continuous data streams. This paper proposes an improved incremental learning algorithm for a large-scale data stream, which is based on SVM (Support Vector Machine) and is named DS-IILS. The DS-IILS takes the load condition of the entire system and the node performance into consideration to improve efficiency. The threshold of the distance to the optimal separating hyperplane is given in the DS-IILS algorithm. The samples of the history sample set and the incremental sample set that are within the scope of the threshold are all reserved. These reserved samples are treated as the training sample set. To design a more accurate classifier, the effects of the data volumes of the history sample set and the incremental sample set are handled by weighted processing. Finally, the algorithm is implemented in a cloud computing system and is applied to study user behaviors. The results of the experiment are provided and compared with other incremental learning algorithms. The results show that the DS-IILS can improve training efficiency and guarantee relatively high classification accuracy at the same time, which is consistent with the theoretical analysis.

Keywords: Data stream, cloud computing, SVM, incremental learning

This work is supported by the National Natural Science Foundation of China (No.61170209, No. 61272508, No. 61202432, No. 61370132 and No.61370092). We express our thanks to the Editors and anonymous reviewers for their detailed comments and valuable contribution to improve our manuscript.

http://dx.doi.org/10.3837/tiis.2014.10.005

1. Introduction

With the advancement of information technology, large amounts of data are being generated from operations in every industry, such as weather monitoring, web logs, phone records, financial transactions, etc. This type of data, often denoted as a data stream [1], is different from traditional data collections due to its high speed, high dimension, real-time and dynamic nature. These data must be processed quickly to get effective information in time to be used for decision making, planning, and researching purposes [1] [2]. For example, we can use data to predict and respond to purchase trends by analyzing customers' behavioral characteristics; weather forecasting accuracy can be improved by obtaining information from the image flow data sent back by satellites; extracting and mining the Internet data stream will help with emergency monitoring and when analyzing public opinion on the Internet. All of these applications have made our daily life more convenient and more intelligent. Cloud computing technology facilitates the data stream process using large-scale clusters and distribution computing, which improves the computing capacity and the transmission efficiency of network services. Using traditional data classification methods to process contemporary data streams is increasingly difficult, because the data form has transformed from static to dynamic. Data are generated continuously over time and the volume of data is infinite, so we cannot store and compute all of the data. Operations performed on a data stream need to be long-term and dynamic, which means that the classification algorithm must be smart enough to learn from its data stream in order to update the classifier and that it must be able to produce highly accurate classifications. Algorithms that can adapt to the features of a data stream and classify it efficiently have become one of the most active research directions in this field.

Among all of the classification technologies, the SVM algorithm has the most advantages over the others when processing a small sample size of non-linear and high-dimensional data. Based on the VC dimension theory and SRM (structural risk minimization inductive principle), which seeks an optimal compromise between empirical and confidence risks, SVM increases the generalization ability of the classifier while maintaining high accuracy [3]. Linearly non-separable problems are resolved using an importing penalty factor and by mapping the problems in a higher dimensional space with the aid of the kernel function. In this way, non-linear problems are converted into linear ones. SVM has better classification effects when handling high dimensional issues because its classifier is simple and the algorithm's computational complexity is not impacted by the dimensions of the data. The support vector set, which is only a small part of the sample data, is introduced to represent the characteristics of the whole data set. So SVM is suitable for incremental learning during which support vector sets of history samples are reserved and to identify the relevant information in the history data.

This paper studies the classification methods of data streams using the SVM theory. In applying this theory, several additional contributions are made: (1) an improved incremental learning SVM algorithm is proposed, referred to as IILS, which is appropriate for data stream classification; (2) by focusing on system load conditions and node performance, an improved incremental learning SVM algorithm, referred to as DS-IILS, is developed to resolve the classification problems in a large-scale data stream; (3) these algorithms are implemented in a cloud computing system and are used to analyze the behaviors of a bank's customers to predict their purchase trends. Meanwhile, the results are compared with other

data classification algorithms to prove the effectiveness of the proposed algorithms.

Section 2 of this paper introduces related work and describes the problems to be studied. In section 3, we present the IILS and DS-IILS algorithms and their computational complexity analyses. The experimental analysis is delineated in section 4. The paper concludes with a presentation of future work that can be carried out as a result of this research.

2. Related work and problem analysis

2.1 Related work

The processing model of a data stream is different from that of a traditional database because of its fast, scalable, and dynamic features. Only part of the data stream can be processed due to its potentially infinite nature. Each record in the data stream corresponds to a time point, called a timestamp. Data in a specific time range are processed according to the user's requirements. The time range is defined as a time window [4]. Data in the time window is part of the data stream, so the result is an approximation within an acceptable error range. There are three commonly used window models: the landmark window model, the sliding window model, and the damped window model [4]. None of these models have an obvious advantage over the others, so different application conditions have to be taken into consideration in order to adopt a suitable window model. The sliding window model [4] is interested in the data in a present window, so old data will be cleared whenever new data arrives. The window will slide on the time axis with the arrival of new data. In this paper, the sliding widow model is adopted due to our application requirements.

There are two primary research directions in data stream classifications and mining algorithms: (1) a single classifier data mining algorithm, in which classification is realized by continuous learning and the updating of classifiers; (2) a composite classifier algorithm, which combines several similar or different classifiers and weights their outcomes linearly to improve accuracy, thereby absorbing the advantages of all classifiers. Focusing on the problems of concept shift, unbalanced data sets, missing data content, and noise, Ghazikhani [5] proposed an integrated classifier in an online neural network to elevate its performance. Ditzler [6] suggested that concept shift and data set unbalancing could be settled jointly. Farid [7] developed new methods for concept shift detecting and self-adaption integrating. Hashemi [8] came up with the FlexDT algorithm to prevent fake concept shifts caused by noise, and thereby improved accuracy. This paper [9] proposed a new SVM model, containing a procedure for data cleaning. The experiment aimed at a large amount of unbalanced data within a data stream.

In other related works, Sun [10] adopted a circulated structure to update multi-classifiers based on the futures of the data stream, and thereby improved its efficiency. Couellan [11] treated the training problem as a common nonlinear optimization problem that had a special decomposition property and adopted incremental gradients to calculate the inner product. The algorithm he developed had better performance when considering prediction accuracy and CPU time. Zheng [12] proposed an online SVM incremental learning algorithm, which consisted of learning prototypes and learning support vectors to resolve learning problems with large-scale data. Hou [13] proposed a RedTree algorithm, in which the classification of multiple data streams was achieved using composite sampling. Cazzolate [14] came up with an incremental decision tree algorithm based on the VFDT (Very Fast Decision Tree) system, which achieved better performance in noise processing and accuracy. Abdallah [15]

introduced a new classification of behavior recognition systems based on computer clusters. It integrated increments from data stream mining and active learning. Dilectin [16] classified the real-time data for tsunami warnings using a KNN (K-Nearest Neighbor) algorithm and dynamically detected data. In the study of human face gender recognition, Yang [17] proposed a new framework for reaching faster face gender recognition using a local ternary pattern and an extreme learning machine. Syed [18] proposed a Batch incremental learning algorithm based on SVM, which could carry out incremental learning effectively using data changes. The training data set used in the network text classification was massive and fast-changing. Ding [19] suggested the FVI-SVM algorithm, which narrowed the volume of the training data set using a KKT (Karush-Kuhn-Tucker) condition and improved classification efficiency. Gonzalez-Mendoza [20] introduced KKT conditions and considered the KKT optimality conditions in order to present a strategy to implement the SVM-QP (quadratic optimization problem based on Support Vector) Machines. Among those studied, the BatchSVM incremental learning algorithm could continually accumulate more support vector sets, but when data endlessly increased, it could bring too great a burden to perform the computations. Besides, the incremental learning algorithm of the KKT conditions filtered more incremental samples during training, so the classification accuracy was lower.

In a weighting method of the SVM algorithm, which treated the gray image as a digital terrain model, Zheng [21] developed an adaptive weighted least squares support vector machine, named the LS-SVM, to iteratively estimate the optimal gray surface underlying the noisy image. The LS-SVM worked on Gaussian noise while the weighted LS-SVM worked on the outliers and the non-Gaussian noise. Xing [22] presented a feature selection and weighted support vector machine (FSWSVM) classifier-based algorithm to detect ships using polarimetric SAR imagery. For the online classification of data streams with imbalanced class distribution, Zhu [23] gave an incremental Linear Proximal Support Vector Machine, named the LPSVM, also called the DCIL-IncLPSVM, which provided robust learning performance in the case of class imbalance. Fergani [24] presented a new way to deal with the class imbalance problem, creating an efficient way of choosing the suitable regularization parameter C of the Soft-Support Vector Machines (C-SVM) method in order to facilitate automatic recognition of activities in a smart home environment. None of these papers considered the case of imbalanced data volumes within a history sample set and an incremental sample set in incremental learning algorithms. During the research reported in this paper, we paid attention to this issue and handled it using weighted processing to improve the classification accuracy.

2.2 Problem analysis

There are several defects in traditional incremental learning algorithms. For example, the standard SVM [9] is not an incremental algorithm. Although its classification accuracy is higher, the problem is that the higher the data volume, the longer the training time. By discarding useless sample sets and keeping useful support vector sets only, the BatchSVM [18] can realize the purpose of a smaller data set volume. But this advantage disappears when it comes to larger amount of support vectors, the subset volume will be larger as a result of continuous iteration and accumulation. Moreover, data stream, which is real-time continuous, will pose a heavy burden on computation. An incremental learning algorithm based on a KKT condition [20] will also decrease the sample amount in incremental samples, which will take part in the next step of training. This lowers the classification accuracy because a lot of incremental samples are filtered at the same time. Moreover, during data stream processing, we take the data of a sliding window as the incremental sample set and

the previous data as the historical sample set. The problem of an unbalanced sample amount emerges when the ratio of the history sample set becomes larger and larger. Data streams all have their own features, so the solutions vary from case to case. For example, when providing user behavior analysis the aim is to analyze the general trend of users, so the major factor affecting the general behavior trend is the scale of the data, that is, slight changes in the users' behaviors cannot affect the general trend. Based on this analysis, we improved the SVM incremental learning algorithm in order to adapt to the data stream features of a data stream used to perform user behavior analysis.

The performance of the SVM incremental learning algorithm depends on reserving important information in the history samples. Properly reserving information will simplify calculation and provide effective information for subsequent training. The key point is the support vector set.

In **Fig. 1**, $f(x)=0$ is the historical optimal separating hyperplane, P_1, P_2, P_3, P_4 are the support vectors in the historical sample set. When new samples A_1, A_2, A_3, A_4, A_5 arrive, $g(x)=0$ is the new optimal separating hyperplane and A_1, A_4, A_5, Q_1, Q_2, Q_3, Q_4 are the new support vectors for the new classifier. Traditional KKT incremental learning finds samples that don't meet the KKT condition and trains them, by which the sample size will be reduced. But in **Fig. 1**, A_5 conforms to the KKT condition, but it is also a support vector of the new classifier. Because A_5 is close to the historical optimal hyperplane, when filtering samples, this kind of data should also be considered. Except for the samples that violate the KKT condition, denoted by the sample point of $y_i f(x_i)<1$, data points like A_5 are reserved as well. Then, we let the condition be all sample points that satisfy $y_i f(x_i)<1+\varepsilon, \varepsilon>0$. ε is imported to constrain samples that are a certain interval apart from the hyperplane, a larger ε means more sample points. When ε is infinite, all samples are included in the new sample set. Also, in researching the reserving samples, we find that Q_1, Q_2, Q_3, Q_4 are changed from non-support vectors to support vectors through incremental learning, so the sample points of the historical sample set that satisfy $y_i f(x_i)<1+\varepsilon, \varepsilon>0$ should also be reserved. Apparently, a larger ε means higher accuracy and a larger sample set, which lead to higher demands in computation. So ε needs to be chosen accordingly. Our research mainly focuses on reserving important classification sample information and discarding useless samples.

In the meantime, our user behavior analysis takes into account the general trend of users, so the number of incremental samples is beneficial for the classifier. Therefore, when historical data and incremental data are weighted separately, our weighing principle is that the weights increase with the volume of data.

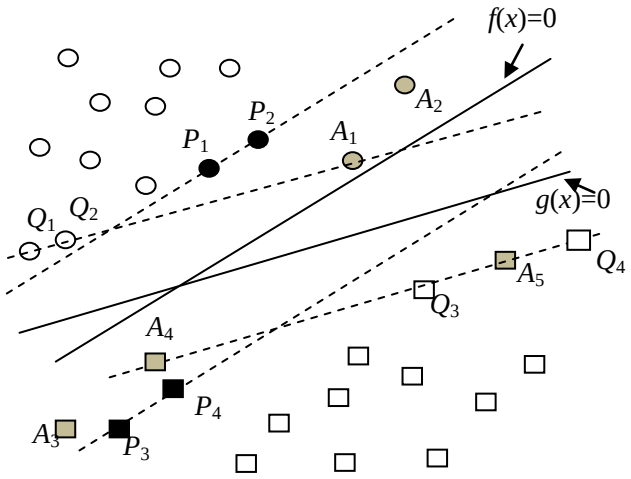

Fig. 1. Change analysis of support vector set after incremental learning

We assume that the historical training set (which number is l) is as formula 1:

$$\{(x_1, y_1), \cdots, (x_l, y_l)\} \in (X \times Y)^l \tag{1}$$
$$x_i \in X = R^n, y_i \in Y = \{1, -1\}, i = 1, \cdots, l.$$

And the incremental sample set (which number is k) is as formula 2:

$$\{(x_{l+1}, y_{l+1}), \cdots, (x_{l+k}, y_{l+k})\} \in (X \times Y)^k \tag{2}$$
$$x_i \in X = R^n, y_i \in Y = \{1, -1\}, i = l+1, \cdots, l+k.$$

We define that the sample weight in $\{(x_1, y_1), \cdots, (x_l, y_l)\} \in (X \times Y)^l$ is denoted by $d_i = \frac{l}{l+k}$, and the weight in $\{(x_{l+1}, y_{l+1}), \cdots, (x_{l+k}, y_{l+k})\} \in (X \times Y)^k$ is denoted by $d_i = \frac{k}{l+k}$.

After increasing the weight of the samples, the quadratic programming problem is changed to:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l+k} \xi_i \tag{3}$$
$$s.t. \quad y_i((w \cdot x_i) + b) \geq d_i(1 - \xi_i), i = 1, \cdots, l+k$$
$$\xi_i \geq 0, i = 1, \cdots, l+k$$
$$d_i = \begin{cases} \frac{l}{l+k}, i = 1, \cdots, l \\ \frac{k}{l+k}, i = l, \cdots, l+k \end{cases}$$

Where $C > 0$ is the penalty factor, so that the transformed dual problem can be described by formula 4, that is, the optimal solution of formula 4 is that of our problem.

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{l+k}\sum_{j=1}^{l+k} y_i y_j \alpha_i \alpha_j (x_i \cdot x_j) - \sum_{i=1}^{l+k} \alpha_i d_i \tag{4}$$
$$s.t. \quad \sum_{i=1}^{l+k} y_i \alpha_i = 0$$
$$0 \leq \alpha_i \leq C, i = 1, \cdots, l+k$$

α is the Lagrange multiplier.

Result $\alpha^* = (\alpha_1^*, \cdots, \alpha_{l+k}^*)^T$ is the optimal solution, so the decision function is $f(x) = \sum_{i=1}^{l} a_i^* y_i (x, x_i) + b^*$, in which $b^* = \frac{1}{2}[(w_* \cdot x_p - d_p) + (w_* \cdot x_q - d_q)]$ and x_p, d_p are any positive class support vector and its weights x_q, d_q are any negative class support vector and its weight.

3. The Improved incremental learning algorithm of SVM

3.1 IILS Algorithm design

Based on the analysis of section 2.2, aiming to resolve the issues of BatchSVM [18] and KKT incremental algorithm [20], which are the problem of reserving valid history data and the problem of imbalanced data volumes in the history sample set and the incremental sample set, a data-stream-oriented improved incremental learning algorithm of SVM, named the IILS, is described below:

Premise：Assuming that A is the initial sample set and B and D are incremental sample sets (we take to two incremental sets for this example). Among them, the amount of samples in A is N_A and N_B and N_D are the amounts of B and D. The data set before the sliding window is the historical sample set and N_0 is the amount of samples. The data set in the current sliding window is the incremental sample set and its samples amount is N_t.

The procedure of IILS is as follows：

1) In the initial state, A is the initial sample set and B is the incremental set, then N_0=N_A, N_t=N_B. By training A, we can acquire the classifier ψ_A, the support vector set SV_A and the decision function $f_A(x)$;

2) With a set threshold of $\varepsilon, \varepsilon > 0$, samples that meet $y_i f_A(x_i) < 1 + \varepsilon, \varepsilon > 0$ in A will be reserved and become the historical outline set, which is denoted by A', while samples that are too far away from the optimal separating hyperplane are discarded;

3) Detect B using ψ_A, if there are no samples that violate the KKT condition, then ψ_A is the classifier after incremental learning, which is denoted by ψ_{A+B}, and the historical outline set is $(A+B)' = A'$, $N_0 = N_A + N_B$;

4) If not all samples of B meet the KKT condition, then samples that are in A' and satisfy $y_i f(x_i) < 1 + \varepsilon, \varepsilon > 0$ in B are trained with a weight to get the classifier ψ_{A+B}, the support vector set SV_{A+B} and the decision function $f_{A+B}(x)$. Samples that satisfy $y_i f_{A+B}(x_i) < 1 + \varepsilon, \varepsilon > 0$ in the training set are reserved, which is denoted by $(A+B)'$, and are then treated as the historical outline set after updating, while samples that are too far away from the optimal separating hyperplane are abandoned;

5) Detect D using ψ_{A+B} and iterate to acquire the new classifier ψ and the decision function $f(x)$ until there are no newly incremental sets.

3.2 DS-IILS Algorithm design

To improve the processing efficiency of a large-scale data stream in the system, i.e. the optimization of the task schedule, we consider the effect of node concurrent access on the system load, which is also one of our optimal strategies. The concurrency access rate of each node A_i, ($1 \le i \le H$) can be calculated using formula 5. Among them, H is the number of nodes

in the cloud. By selecting nodes that have lower concurrent access rates as work nodes, the training efficiency of the whole system is improved.

$$A_i = \frac{Task_i}{SumTask_i} \tag{5}$$

$Task_i$ is amount of tasking executing in ith node and $SumTask_i$ is the max task number that can be executed in it. If there are H nodes in the cloud, the overall concurrent access rate of the system is described by formula 6:

$$C_A = \sum_{i=1}^{H} A_i \Big/ H \tag{6}$$

The load threshold L is defined to denote C_A, then $L=C_A$. If $0 \le A_i \le L$, then the ith node belongs to the light-load node set L_{light}; otherwise it belongs to the heavy-load node set L_{heavy}. Selected nodes must be nodes of L_{light} so that the system load balance is guaranteed and the processing efficiency is improved.

Based on previous considerations, the proposed improved incremental learning algorithm of SVM focusing on the large-scale data streams, named the DS-IILS, is described below:

First: data stream preprocessing

(1) The data stream is divided into blocks with the same size of xMB and they are sorted by arrival time;

(2) The blocks amount is denoted by E;

Second: analysis of system load condition

(1) Calculate the concurrent access rate of each node A_i using formula 5;

(2) Calculate the concurrent access rate of the whole system C_A using formula 6 and define the load threshold $L=C_A$;

(3) If $0 \le A_i \le L$, then the ith node belongs to L_{light}, if not then it belongs to L_{heavy}. The number of nodes in L_{light} is M;

(4) Sort nodes of L_{light} in ascending order by their concurrent access rate.

Third: data stream parallel processing

(1) If $E \le M$, then data blocks are put into sorted nodes by their arriving order. All nodes parallel process the blocks which are assigned to them; that is, each node will call IILS and execute incremental learning internally simultaneously with all other nodes. Then, the decision function of each node is weighted and voted according to the amount of samples that have been processed. The final decision function $f(x)$ is determined and the algorithm ends.

(2) If $E>M$, then the previous M data blocks are put into sorted nodes by their arriving order. Meanwhile, all nodes parallel process the blocks which are assigned to them; that is, each node will call IILS and execute incremental learning internally simultaneously with all other nodes. The decision function and the amount of processed samples are reserved. Make $E=E-M$ and return to the second step.

3.3 Computation complexity analyses of IILS and DS-IILS

The computational complexity of the SVM is affected by the dimension of samples n, the scale of sample set l, the amount of support vectors N_{SV}, and their distribution. Commonly, N_{SV} is much smaller than l, that is $(N_{SV}/l) << 1$, so the computational complexity of SVM is $O(N_{SV}^3 + lN_{SV}^2 + nN_{SV})$. Assume the scale of the newly incremental sample set is k and

$0 < k/l < 1$. In traditional SVM algorithms, the newly incremental samples are added to the historical samples to be retrained and the scale of the sample set is $l(1+k/l)$, so their computational complexity is $O(N_{SV}^3 + l(1+k/l)N_{SV}^2 + nN_{SV})$. $N_{SV}^{'}$, which is the scale of the sample set including the samples of the history sample set and the newly incremental sample set that are within the scope of the threshold, will be much smaller than $l(1+k/l)$ after adopting IILS; that is, $N_{SV}^{'}/(l+k) << 1$. Generally, $N_{SV}^{'}/N_{SV} \approx 1$, so the computational complexity of IILS is $O(N_{SV}^3 + nN_{SV}^2)$, which is less than the complexity of other algorithms.

The core of the DS-IILS is the IILS, but the DS-IILS also considers the system load and adopts parallel optimization, so the run time is greatly reduced and the work efficiency is significantly improved.

4. Experiment

4.1 Simulation environment and data set

The campus cloud of USTB (University of Science and Technology Beijing), in which Hadoop was adopted as the underlying storage, MapReduce as the upper parallel computing framework and virtualization technology was also realized, served as the experimental platform. Our evaluations of the algorithms are performed within this campus cloud. For now, it consists of 10 data nodes with different computer performances. Their configurations are shown in **Table 1**. Algorithms are evaluated based on two aspects: (1) training time, which reflects the algorithm's efficiency and (2) classification accuracy, which represents the algorithm's learning abilities and classification accuracies. Meanwhile, to ensure the accuracy and effectiveness of the experiment's results, we adopt Java for all algorithms, each experiment is run 50 times under the same condition, and the average results are reported.

We adopt Clementine12 to preprocess data and analyze the importance of variables. Our algorithm is realized based on the libSVM [25]. The data set is pulled from the Bank Marketing data sets [26] in the UCI database[1]. The data are generated by marketing behaviors of financial institutions in Portugal which try to predict customers' purchase behaviors by classification. There are 17 attributes in the data set, among which IDs ranging from 1 to 9 are customer attributes, including age, job, and education, etc., and IDs ranging from 9 to 16 are customers' contact information, and ID=17 is the target attribute in which true is purchase behavior and false is non-purchase behavior.

Table 1. Configuration of cloud nodes

	D_1（2）	D_2（2）	D_3（2）	D_4（2）	D_5（2）
Configuration	1core/ 2G/ 120G	1core/ 4G/ 240G	2core/ 8G/ 480G	4core/ 16G/ 960G	8core/ 32G/ 1T
Operating System	Windows XP	Windows XP	Windows XP	Windows XP	Windows XP
Java Version	1.7.0_01	1.7.0_01	1.7.0_01	1.7.0_01	1.7.0_01
Software Environment	Eclipse, Clementine12	Eclipse, Clementine12	Eclipse, Clementine12	Eclipse, Clementine12	Eclipse, Clementine12

[1] The UCI database is a standard test dataset in the field of Machine Learning which is raised by the University of CaliforniaIrvine, Irvine. Currently, there are 294 data sets in it. http://archive.ics.uci.edu/ml/

4.2 Selection of kernel function and penalty factor

In these experiments, we mainly compare the RBF kernel function with a polynomial kernel function. The numbers of samples are 4505, 9025, 13724, and 17960 respectively. And the values of the penalty factor *C* are 0.1, 1, 10, and 100. The results are shown in **Table 2**.

There are several factors to consider when choosing a kernel function. Although the training time of the polynomial kernel function is slightly shorter than that of the RBF kernel function, the overall training accuracy of the classifier of the RBF kernel function is higher than that of the polynomial kernel function. Furthermore, it is generally known that if the parameters of the kernel function are greater, then the stronger classification result depends on the parameters and it will consume too much time selecting the parameters. The parameter number of the RBF kernel function is less than that of the polynomial kernel function; therefore, we selected the RBF kernel function for this experiment.

Moreover, considering the selection of penalty factor *C*, we can see in table 2 that when *C* is only 100, the training time is obviously longer, but in other cases the training time is stable; so, the value of *C* will be selected from 0.1, 1, and 10. According to the classification accuracy, when *C* is 10, the accuracy is higher and steadier in different samples; so, we set the value of the penalty factor *C* at 10 for this experiment.

Table 2. Contrast experiment of kernel function and penalty factor

RBF kernel function				**Polynomial（poly）kernel function**			
Sample size	*C*	**Training time**(s)	**Accuracy** (%)	**Sample size**	*C*	**Training time**(s)	**Accuracy** (%)
4505	0.1	3.750	87.86	4505	0.1	3.344	87.86
	1	3.484	88.01		1	3.172	87.68
	10	3.954	89.03		10	3.516	89.12
	100	8.125	89.01		100	9.110	88.75
9025	0.1	15.984	87.86	9025	0.1	14.672	87.86
	1	14.328	88.87		1	13.031	88.66
	10	15.468	89.03		10	13.938	89.06
	100	33.719	88.63		100	36.234	88.77
13724	0.1	41.281	87.68	13724	0.1	37.766	87.86
	1	36.062	88.97		1	33.313	88.77
	10	38.375	89.50		10	35.157	89.68
	100	78.328	90.05		100	82.813	89.92
17960	0.1	75.891	88.64	17960	0.1	68.219	87.86
	1	66.094	89.07		1	61.875	88.79
	10	69.219	88.57		10	65.016	88.44
	100	143.047	88.28		100	148.172	87.81

4.3 Parameter selection in improved incremental algorithm

Historical samples that satisfy $y_i f(x_i) < 1+\varepsilon, \varepsilon > 0$ have to be reserved in the DS-IILS. A properly chosen ε will lead to better classification performance, so we take contrast experiments on ε. The size of the initial sample set is 5000 and the same amount is in incremental sample sets 1 through 3. Based on the analysis in section 4.2, the kernel function we adopt is the RBF and the penalty factor *C*=10. Select the value of ε from among 0, 0.2,

0.4, 0.6, and 0.8. We conduct the experiment 10 times in the same conditions, and the average results are reported in **Fig. 2** and **Fig. 3**. **Fig. 2** shows its effect on training time and **Fig. 3** depicts its effect on accuracy.

According to the results, we can conclude that the larger ε will lead to more training time and higher accuracy. 0.4 is chosen to be the value of ε for the comparison.

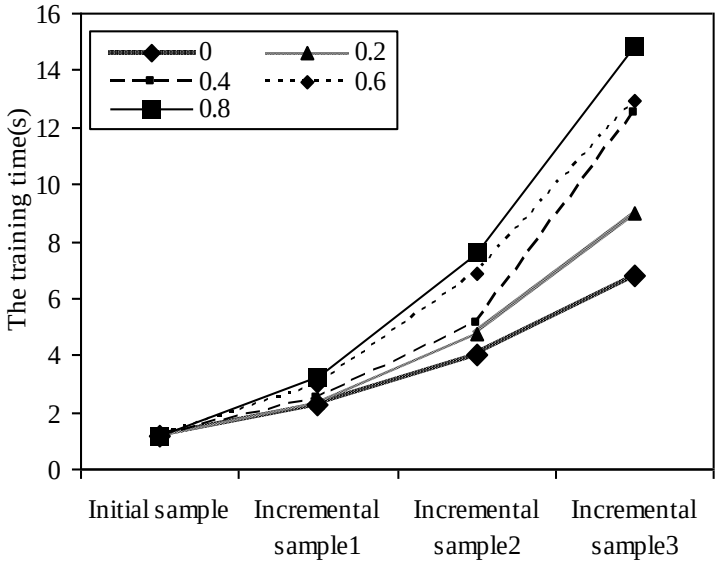

Fig. 2. Effect of ε on training time

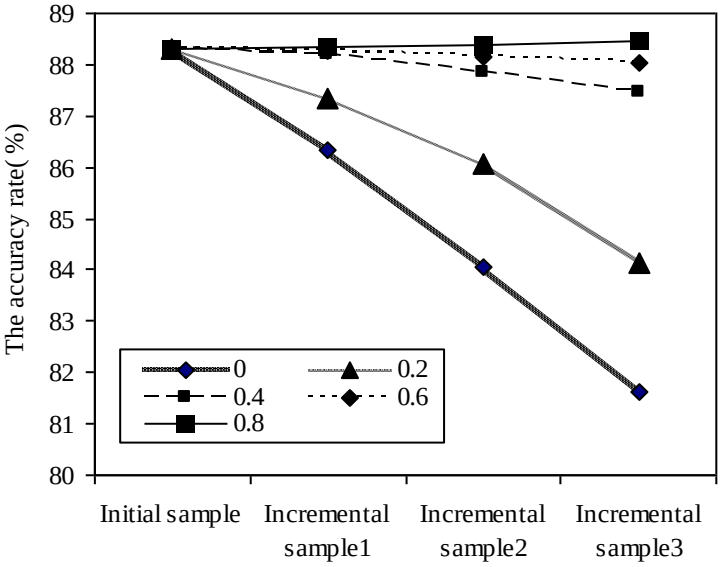

Fig. 3. Effect of ε on accuracy

4.4 Experimental comparisons

The IILS algorithm is the core of the DS-IILS algorithm in this experiment, i.e., the IILS is a subfunction of the DS-IILS and an exception of the DS-IILS when the number of cloud nodes is one. In the experimental comparisons explored during this research, we compared

the DS-IILS with the standard SVM [9], the BatchSVM incremental algorithm [18], and the KKT incremental algorithm [20], thereby excluding the IILS. Again, the focus of our comparisons includes training time and accuracy in order to determine the effectiveness of the DS-IILS algorithm. In **Fig. 4** and **5**, the sizes of the initial sample and the incremental sample sets 1 through 3 are all 5000; the size of the testing sample set is 20000. The kernel function is RBF and the penalty factor C is 10. ε is set at 0.4. The size of data block x is 32MB. To assure the accuracy of experiment results, we conduct the experiment 50 times under the same conditions and the average results are reported in **Fig. 4** and **Fig. 5**. Training time is compared in **Fig. 4** and comparison of accuracy is in **Fig. 5**. From the results, we can see that, in general, the incremental learning algorithm can improve efficiency to a large extent with only a little decline in accuracy, so the incremental learning algorithm is effective. This is consistent with the complexity analyses of the algorithms in section 3.3. Also, the accuracy of the KKT incremental algorithm decreases significantly when a lot of data emerge because it reserves fewer samples. Although the training time of the DS-IILS is slightly longer than the KKT incremental algorithm, the accuracy of the DS-IILS is obviously improved. Besides, the DS-IILS has a higher efficiency when compared with the BatchSVM algorithm, especially when data is continuously increasing because it will not accumulate a large amount of samples. So the performance of the DS-IILS is superior.

Next, the sample sizes are enlarged to prove its high efficiency in processing a large-scale data streams. The sample sizes of the initial sample set and the incremental sample sets 1 through 3 are 1 million and the testing sample size is 100,000. **Fig. 4** shows that DS-IILS is apparently better than the standard SVM algorithm, so we only compare it with the KKT algorithm and the BatchSVM algorithm. The results are shown in **Fig. 6**, from which we can conclude that the system training time of the DS-IILS is much shorter than that of the other algorithms. This is because the DS-IILS considers not only the system load but also the node performance. When scheduling tasks, it can first select the better performing node to process the task.

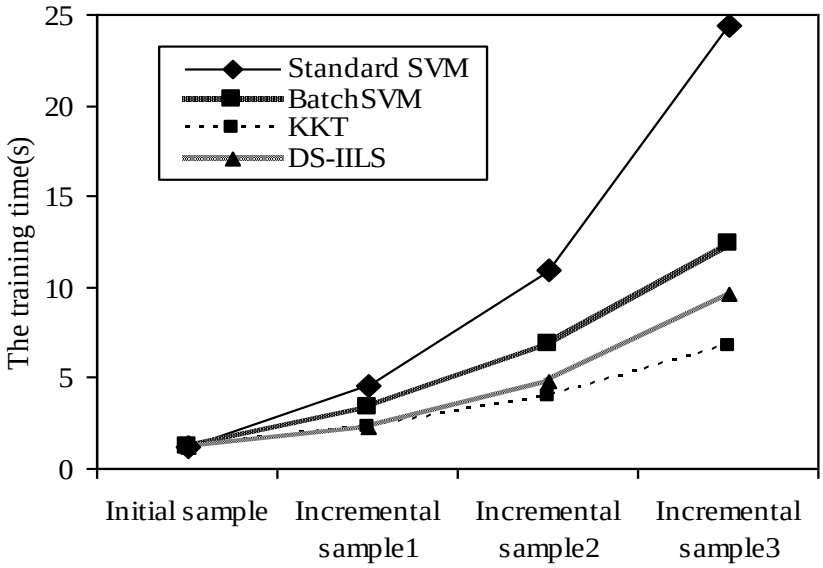

Fig. 4. Comparison of training time with 4 algorithms

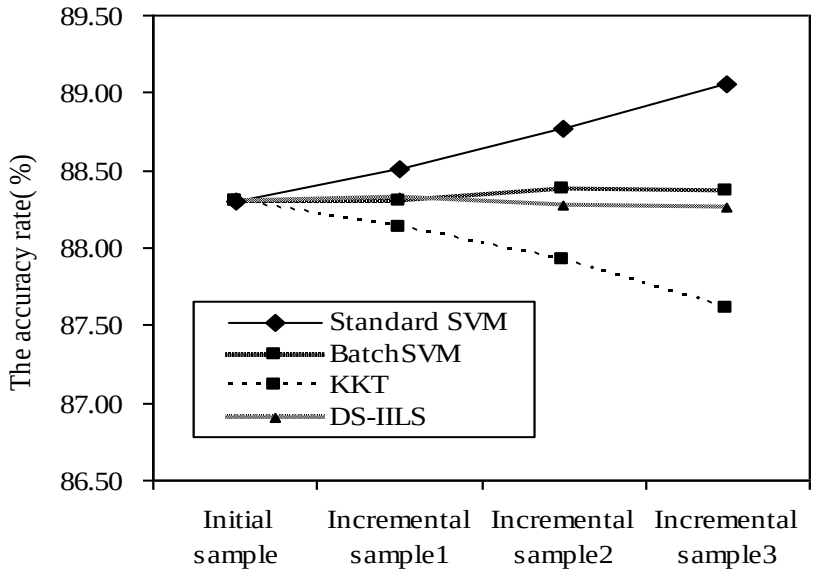

Fig. 5. Comparison of accuracy with 4 algorithms

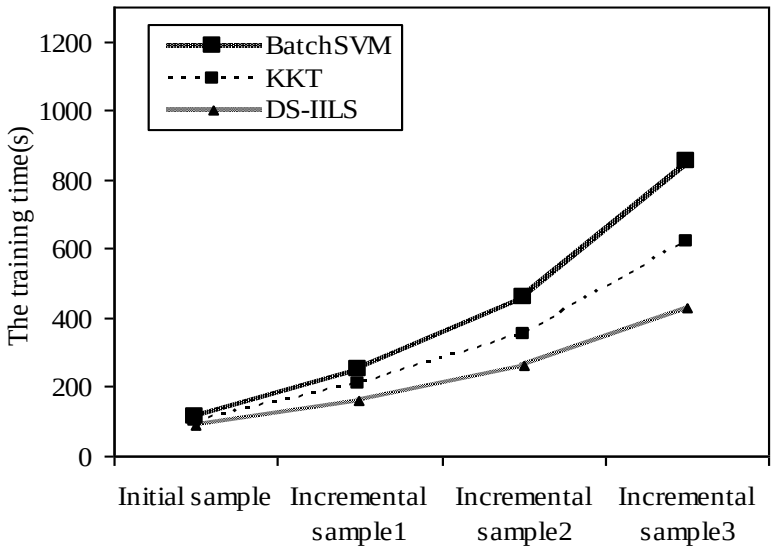

Fig. 6. Comparison of training time on a large-scale data stream with 3 algorithms

5. Conclusion

This paper reports on our efforts to improve the SVM incremental learning algorithm as it applies to the classification methods of data streams, specifically to adapt it to the features of a large-scale data stream used for customer behavior analysis. First, we proposed an improved incremental learning algorithm based on the SVM theory, in which the threshold of the distance from the optimal separating hyperplane is defined. Historical samples and incremental samples that satisfy the limits of the threshold are reserved to construct the training sample set. The ratio of the historical sample set and the incremental sample set is also considered and processed using weighting. Second, by taking the load condition and

node performances into consideration, we took advantage of cloud computing capabilities to improve the incremental learning algorithm based on the SVM theory for use with a large-scale data stream, thereby improving the training efficiency and the classification accuracy of the system. Finally, the proposed algorithm is implemented in the cloud platform and compared with the standard SVM algorithm, the BatchSVM incremental algorithm, and the KKT incremental learning algorithm in the process of analyzing customer behaviors. The experiment results show that the DS-IILS algorithm has a higher training efficiency and classification accuracy for a large-scale data stream, which is consistent with the theoretical analysis. In this paper, we research two-classification incremental learning algorithms based on SVM. Our future work will address the problems with multi-classification incremental learning algorithms.

References

[1] W. Wu and L. Gruenwald, "Research issues in mining multiple data streams," in *Proc. of the First International Workshop on Novel Data Stream Pattern Mining Techniques,ACM*, pp.56-60, July 25, 2010. Article (CrossRef Link).

[2] Y. B. Yuan, J. Lu and F. L. Cao, "Baby formula classification based on forth order polynomial smoothing support vector machine," in *Proc. of International Conference on Machine Learning and Cybernetics*, pp. 685-691, July 10-13, 2011. Article (CrossRef Link).

[3] S. Pang, L. Zhu, G. Chen, A. Sarrafzadeh, T. Ban and D. Inoue, "Dynamic class imbalance learning for incremental LPSVM," *Neural Networks*, vol. 44, pp.87-100, August, 2013. Article (CrossRef Link).

[4] W. Ding, Y. Han, J. Wang and Z. Zhao, "Space reduction for extreme aggregation of data stream over time-based sliding window," in *Proc. of 2012 IEEE 5th International Conference on Cloud Computing*, pp. 1002-1003, June 24-29, 2012. Article (CrossRef Link).

[5] A. Ghazikhani, H. S. Yazdi and R.Monsefi, "Class imbalance handling using wrapper-based random oversampling," in *Proc. of 2012 20th Iranian Conference on Electrical Engineering (ICEE 2012)*, pp. 611-616, May 15-17, 2012. Article (CrossRef Link).

[6] G. Ditzler and R. Polikar, "Incremental learning of concept drift from streaming imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 10, pp. 2283-2301, October, 2013. Article (CrossRef Link).

[7] D. M. Farid , L. Zhang, A. Hossain, C. M. Rahman, R. Strachan, G. Sexton and K. Dahal, "An adaptive ensemble classifier for mining concept drifting data streams," *Expert Systems with Applications*, vol. 40, no. 15, pp. 5895-5906, November, 2013. Article (CrossRef Link).

[8] S. Hashemi and Y. Yang, "Flexible decision tree for data stream classification in the presence of concept change, noise and missing values," *Data Mining and Knowledge Discove*ry, vol. 19, no. 1, pp: 95-131, August, 2009. Article (CrossRef Link).

[9] Y. Tang, Y. Q. Zhang, N. V. Chawla and S. Krasser, "SVMs modeling for highly imbalanced classification," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 39, no. 1, pp. 281-288, February, 2009. Article (CrossRef Link).

[10] N. Sun and Y. Guo, "Model on data stream classification with incremental learning," *Computer Engineering and Design*, vol. 33, no. 11, pp. 4225-4229, November, 2012. Article (CrossRef Link).

[11] N. P. Couellan and T. B. Trafalis, "On-line SVM learning via an incremental primal-dual technique," *Optimization Method Software*, vol. 28, no. 2, pp. 256-275, February, 2013. Article (CrossRef Link).

[12] J. Zheng, F. Shen, H. Fan and J. Zhao, "An online incremental learning support vector machine for large-scale data," *Neural Computing Applications*, vol. 22, no. 5, pp. 1023-1035, May, 2013. Article (CrossRef Link).

[13] W. Hou, B. Yang, C. Wuc and Z. Zhou, "RedTrees: a relational decision tree algorithm in streams," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6265-6269, September, 2010.

Article (CrossRef Link).
[14] M. T. Cazzolato, M. X. Ribeiro, C. Yaguinuma and M. T. P.Santos, "A statistical decision tree algorithm for data stream classification," in *Proc. of the 15th International Conference on Enterprise Information Systems*, pp. 217-223, July 4-July 7, 2013. Article (CrossRef Link).
[15] Z. S. Abdallah, M. M. Gaber, B. Srinivasan and S. Krishnaswamy, "StreamAR: incremental and active learning with evolving sensory data for activity recognition," in *Proc. of In Tools with Artificial Intelligence (ICTAI), 2012 IEEE 24th International Conference on*, pp. 1163-1170, November 7-9, 2012. Article (CrossRef Link).
[16] B. V. Mercy, "Classification and dynamic class detection of real time data for tsunami warning system," in *Proc. of In Recent Advances in Computing and Software Systems (RACSS), 2012 International Conference on*, pp. 124-129, April 25-27, 2012. Article (CrossRef Link).
[17] J. Yang, Y. Jiao, N. Xiong and D. Park, "Fast face gender recognition by using local ternary pattern and extreme learning machine," *KSII Transactions on Internet and Information Systems*, vol.7, no.7, pp.1705-1720, July, 2013. Article (CrossRef Link).
[18] N. A. Syed, S. Huan, L. Kah and K. Sung, "Incremental learning with support vector machines," in *Proc. of the Workshop on Support Vector Machines at the international Joint Conference on Artificial Intelligence (IJCAI 1999)*, pp.352-356, July 31-August 6, 1999. http://citeseer.ist.psu.edu/viewdoc/download?doi=10.1.1.46.6367&rep=rep1&type=pdf.
[19] W. Ding and A. Xue, "Fast incremental learning SVM for Web text classification," *Application Research of Computers*, vol. 29, no. 4, pp. 1275-1278, April 2012. Article (CrossRef Link).
[20] M. Gonzalez-Mendoza and R.E.I. Orozco, "Quadratic optimization fine tuning for the support vector machines learning phase," *Expert Systems with Applications*, vol. 41, no. 3, pp. 886-892. January, 2014. Article (CrossRef Link).
[21] S. Zheng, C. Yang, E. A. Hendriks and X. Wang, "Adaptive weighted least squares SVM based snowing model for image denoising," *International Journal of Wavelets Multiresolution and Information Processing*, vol. 11, no. 6, pp. 1-25, November, 2013. Article (CrossRef Link).
[22] X. Xing, K. Ji, H. Zou and J. Sun, "Feature selection and weighted SVM classifier-based ship detection in PolSAR imagery," *International Journal of Remote Sensing*, vol. 34, no. 22, pp. 7925-7944, September, 2013. Article (CrossRef Link).
[23] L. Zhu, S. Pang, G. Chen and A. Sarrafzadeh, "Class imbalance robust incremental LPSVM for data streams learning," in *Proc. of In Neural Networks (IJCNN), The 2012 International Joint Conference on*, pp. 1-8, June 10-15, 2012. Article (CrossRef Link).
[24] B. Fergani and L. Clavier, "Importance-weighted the imbalanced data for C-SVM classifier to human activity recognition," in *Proc. of In Systems, Signal Processing and their Applications (WoSSPA), 2013 8th International Workshop on*, pp. 330-335, May12-15, 2013. Article (CrossRef Link).
[25] C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1-27, April, 2011. Article (CrossRef Link).
[26] S. Moro, P. Cortez and P. Rita. "A Data-Driven Approach to Predict the Success of Bank Telemarketing," *Decision Support Systems*, vol. 62, pp: 22-31, June, 2014. Article (CrossRef Link).

Ning Wang received the B.E. degree in computer science and technology from Shandong University of Technology, China, in 2003, and the M.S. degree in computer software and theory from Yunnan Normal University, China, in 2006. She is currently working toward the Ph.D. degree with School of Computer and Communication Engineering, University of Science and Technology Beijing, China. Her current research interests include cloud computing and data mining. Email: wangning200658@126.com.

Yang Yang received his Ph.D. degree from University of Science and Technology in Lille, Lille, France in 1988. Currently, he is a professor and a Ph. D. supervisor in School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China. His research interests include cloud computing, intelligence control and multimedia communication. Email: yyang@ustb.edu.cn.

Liyuan Feng received the B.S. degree from University of Science and Technology Beijing, China, in 2011, and the M.E. degree from University of Science and Technology Beijing, China, in 2014. Her current research interests are big data, cloud computing and data stream analysis. Email: fengliyuar@163.com.

Zhenqiang Mi received B.S. in automation and Ph.D. in communication engineering, both from School of Information Engineering, University of Science and Technology Beijing, in the year 2006 and 2011, respectively. From 2011, he has been an assistant professor with the school of computer and communication engineering, University of Science and Technology Beijing. His research interest includes mobile ad hoc networks and cloud computing in mobile environments. Email: mizq@ustb.edu.cn.

Kun Meng received his Ph.D. degree from University of Science and Technology Beijing, Beijing, China. Now, he is a Postdoctor Fellow of the Department of Computer Science and Technology, Tsinghua University. His research interests include QoS of computer networks, performance evaluation, network security analysis, service computing and stochastic models. Email: mengkun1024@163.com.