2014

ISSN : 0974 - 7435 Volume 10 Issue 13

BioTechnology

An Indian Journal

FULL PAPER

BTAIJ, 10(13), 2014 [7160-7170]

Large-scale data classification based on clustering feature tree decomposition

Yanfeng Li
Heilongjiang University, Department of Information Science and Technology,
Harbin 150080, (CHINA)
E-mail : By_beyond@126.com

ABSTRACT

When the scale of training dataset is large, the demand for computing resource of traditional classifiers will increase fast. So we need to expand SVM algorithms to large-scale dataset. With the analysis on the development and direction of semi-supervised algorithms at home and abroad, this paper introduces clustering feature tree to organize large-scale data using local learning strategy. First, based on the idea of local learning, we use CF tree to organize and separate the samples into a series of local sub-set, to divide original problem into limited small-scale sub-problems; Next, we propose the computing method to improve the Euclidean distance of CF tree, to measure the distance between test samples and multiple local classifiers, and to select the closest classifier for testing; Finally, SVM is used to construct multiple local classifiers for the local labeled clusters. Then these local classifiers are combined to a global classifier to acquire an integrated classification model. Several groups of large-scale data experiments show that the improved algorithm increases the training speed and test speed, with higher test accuracy.

KEYWORDS

CF tree; SVM classifier; Clustering; Local learning; Large-scale data.

INTRODUCTION

Large data brings unprecedented challenges to machine learning. The complexity of traditional machine learning algorithms is more than first power generally, and it requires enough memories to load dataset. So the computing resource is deficient when processing large-scale data for conventional algorithms[1-3]. Recently, many efficient training methods for large-scale data are proposed. They are summarized to be two kinds of technologies: the first is numerical technique, that is, proposing novel optimal model and corresponding training algorithms adapted for large-scale training; the second is data reduction technique, which focuses on reduce the scale of training. The latter is based on the idea of local learning: it separates the training samples to a limited number of local parts and processes large-scale problems with training of each local part[4,5]. The key of such algorithms is separating the training samples. For support vector machine (SVM) algorithms, since it is a convex quadratic programming problem[6], we can ensure the existence and uniqueness of solution during the solving process. But its time complexity is the cube of the number of training samples, and its space complexity is the square of the number of training samples. When processing large-scale dataset, the complexity in space and time are too high for SVM. With the scale increasing, the problems become more prominent. So it is a significant issue on how to apply SVM algorithms to the study of large-scale dataset.

Graf proposed Cascade SVM in 2005[7]. He separated the training set to several sub-set. These sub-sets are trained separately to acquire the supported vectors. Then the supported vectors are merged into several sub-sets again, for the next round of training until the classifier of the whole training set is obtained. Zhang further proposed SVM-KNN algorithm in reference[8]. In 2010, Cheng etc., proposed ProfileSVM, which separated the training set into multiple clusters. Linear SVM classifier will be established for each cluster. In the same year, Chang[9] proposed Decision-tree SVM (DTSVM) based on decision tree decomposition. DTSVM adopts the training samples to establish a decision tree and assign the samples to leaf node. Then the sample of each leaf node will generate a sub-classifier by study. At last these sub-classifiers are integrated to a total classifier.

By study we find most of existing semi-supervised classification algorithms have higher time complexity. Their training time will get sharp increment with the increase of samples, to be hared for application on large-scale training. So we choose data reduction to study large scale semi-supervised classification algorithms. It can not only make data compression and storage, which saves the overhead of memory, but also it can establish tree by one time of scanning, which is easy to separate the dataset with high efficiency. Therefore, this paper uses clustering feature tree (CF Tree) to organize large-scale unlabeled samples. With the idea of local learning, we adopt improved CF tree to organize and separate these samples to a series of local sub-set. Fist we study the semi-supervised classification framework based on CF tree decomposition and local learning. The framework uses the characteristics of CF tree to separate quantities of standard samples to a series of local areas. According to single training of the data in local area, we estimate the mark of unlabeled samples in this local area. Then we propose a clustering algorithm based on CF tree decomposition. It uses dataset to construct new CF tree to form micro-clusters, which also improved the computing equations of Euclidean distance. Birch algorithm is use to cluster for the micro-clusters. At last we establish SVM classifier for each data cluster whose size and scale is very suitable. In this way, we get multiple local classifiers and they are integrated into a global classifier by the improved CF tree. In the remainder of this work, we first discuss the local learning method with CF tree and explain the principle idea in section 2; Then an improved classifier for large-scale data is proposed in section 3; Section 4 verifies the performance of improved scheme by simulations; In the last section we conclude this paper and provide working emphasis in future.

LOCAL LEARNING OF CF TREE DECOMPOSITION

Each node of CF tree denotes a local area[10]. We can acquire given scale of local area samples by preorder traversal of whole CF tree. So according to the scale that is beard by local learning methods, the training scale threshold is set to control automatically the decomposition process of CF tree. The data space is decomposed into a series of local areas and the data of each area will perform local learning[11]. Assuming the time complexity of local learning is $O(n^{\lambda})$, n is the scale of dataset, the training time complexity of each local parts is $O(\tau^{\lambda})$. Then total time complexity is $O(n/\tau \times \tau^{\lambda}) = O(n\tau^{\lambda-1})$. If $\lambda > 1$, compared to global learning, the efficiency of local learning can be increased $(n/\tau)^{\lambda-1}$ times; while the space complexity is reduced to $O(n^{\gamma})$ from $O(\tau^{\gamma})$. Figure 1 depicts the idea of semi-semi-supervised local learning. It can be seen 4 local parts need to train the classified planes in all 5 local parts. The other one does not need to be trained because it only contains one kind of sample points. From the effect of classification, 4 decision planes can be integrated to one decision plane approaching to global one, which shows that local learning can reduce total training scale when ensuring the classification accuracy.

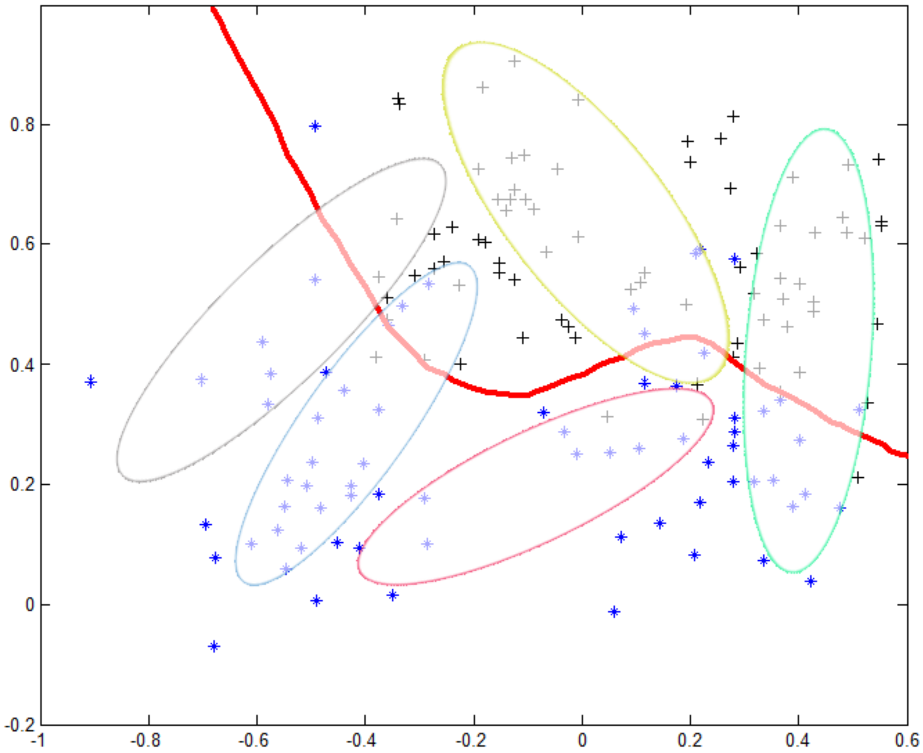

Figure 1 : Semi-supervised local learning in artificial dataset.

Generally the idea of training model in supervised learning is inputting all acquired training samples to learning machine for study, and it will get a classifier model finally. When the number of training samples is small, this method has high feasibility, accuracy and generalization. But for large-scale data, it tends to cause long training time and low classification accuracy due to internal mechanism of the model, which restricts further implementation of this method. That is also the problem in SVM[12,13]. For such SVM problems, we will provide a data separation algorithm based on local learning to construct multiple local classifiers. Then they are integrated to a global classifier by CF tree. The idea of local learning is different from integrated learning. The former does not need to integrate acquired classifiers into one, but the latter does. For a new testing point, we can determine corresponding cluster belonging to the separation by some mechanisms or means. Then the testing point is substituted in to corresponding classifier of this data cluster. From the judging process of testing point, we find local learning method obviously save the training time and training speed, which also ensures the model speed and generalization.

To decrease the computing complexity in SVM global classifier, under the idea of local classification, we propose a CF Decomposition Based SVM classifying algorithm (CFD-SVM) in the following section. Both CFD-SVM and BC-SVM[14] adopt the data organization mode of clustering algorithm BRICH[15,16]. CFD-SVM uses the training samples to establish a labeled CF tree, with supervised methods and unsupervised clustering. The scheme in training is based on local learning; BC-SVM establishes CF tree for different classifications with unsupervised clustering. The scheme in training is based on global learning. Compared to DTSVM, both of two algorithms use tree structure as the technique for data reduction. The difference is: CFD-SVM needs only once scanning of data when establishing trees, and it can choose to compress and store data to adapt available memories, according to memory status; while DTSVM need multiple scanning of data when establishing trees and it can not make compression or storage of data.

DATA CLASSIFIER BASED ON CF TREE AND SVM

Construtuion of local sub-set

Based on the idea of local learning, this paper adopts clustering feature tree to organize and separate the samples to a series of local sub-set, which separates original large-scale classification to limited small-scale dub-problems. So traditional semi-supervised classifying algorithms are feasible due to the decrease of learning scale. It also helps to raise the learning speed. In simple terms, we can use all the given samples to establish a CF tree, labeled and unlabeled samples included. The learning begins with traditional semi-supervised classifying algorithms fir the sub-trees. The sub-tree is selected by the method of threshold setting: assuming β is scale threshold of sub-tree, if the total number of samples in some sub-tree is less than β, we think this sub-tree is suitable for learning. In this paper, we use the unlabeled samples in each sub-set to combine all the given labeled samples for learning, so as to ensure each local problem is semi-supervised one.

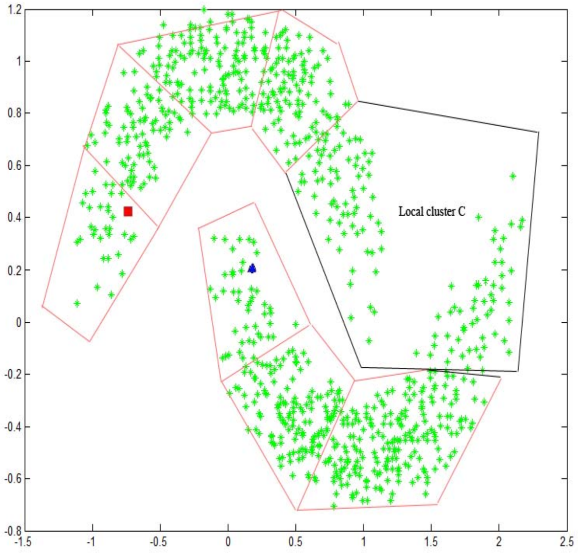

Figure 2 : Schematic diagram of local learning.

In the 2moos problem as shown in Figure 2, top half-moon is positive class samples and the below is negative class samples. Total sample is separated into a number if local sub-sets. Red square and blue triangle respectively labeled points of above classes. We first consider using these two labeled points and the unlabeled points in each polygon for learning, to reduce the learning scale. If only so it will cause another problem. Since semi-supervised is generally based on clustering and assuming manifold, and the samples with close distance are classified as one class. So there exists such case that the labeled points which are closer to some local unlabeled sample set are not correct. As shown in Figure 2, C is a local unlabeled sample set. When we use give two labeled samples and unlabeled samples in C to study, the samples belong to positive class are marked as negative cone, because C has closer distance to the triangle. It is believed that the error is caused by destroyed global structure of samples, as is not hoped by us.

To avoid or slow down the influence on classification accuracy by global structure damage, we can use some key points to keep global structure. Then they can help to broadcast labels of correct labeled samples to unlabeled samples, during the process of local learning, as is shown in Figure 3. We adopt the method in reference[17] when selecting the key points, which are calculated by the clustering center of some CF-layer item in CF trees. Though the method is convenient, there needs more key points to keep better accuracy, which leads to large scale of learning. Thus, based on the idea of AGR and PVM[18], we adopt k-means clustering center as anchor and original vectors to acquire better effects. Different from the algorithm in[19], the underlying algorithm for local learning is similar to AGR. We first learn the class label of the key points. Then their similarity is used to calculate the sample label for fast learning speed. Using k-means can acquire the key points and control their numbers effectively. It also causes small-scale local learning. So the algorithms like LGC and LP can be used as underlying algorithms, which can make full use of the connection information among the samples to keep accuracy.

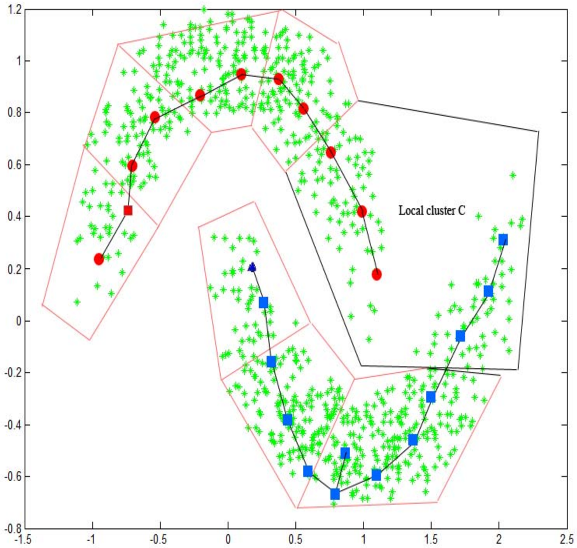

Figure 2: Label broadcasting with key points.

When the sample scale is too large too produce higher price of k-means, CF tree will play another role. In BIRCH algorithm, it adopts CF tree to compress the samples. Then the compressed samples, instead of original samples are clustered, which improves the efficiency of clustering. In the next we will obtain original clustering center set from some layer of CF tree. Then the center set is re-clustered to lower the price of k-means, before the key points are acquired.

CF tree clustering

The principle idea of our improvement is: based on the data structure of CF tree, all the data point are stored in this CF tree. When the tree is completed, we use the leaf nodes of CF tree as input and its clustering feature compressed the input data. So quantities of closed data points will be stored in one clustering feature. The improved algorithm will automatically filter abnormal points during the process of establishing trees. Once the process is over, abnormal points are reassigned to suitable positions of the tree.

Since CF tree is only suitable for the clustering problems of clusters with little difference in volume, micro-cluster**[20]** is used by us. First, micro-cluster is formed by the data point. Different from Brich, it will not upgrade the size of threshold to form larger cluster. Instead, the formed micro-clusters are taken as input to perform clustering. Therefore, it can solve the clustering problems for the clusters with big difference in volume. So it demands that the threshold of CF tree T be well controlled, then the formed micro-clusters will not too small or too large. Based on the setting methods for threshold T in reference**[21]**, we provide a little modification to make it adapt to the algorithm of this paper:

Randomly select N pairs of data object in the dataset

Compute Euclidean distance of each pair of object

Compute the expectations EX and variance DX of the distance of these objects

Set the value of threshold T as $P(EX + DX)$. P is an adjustment factor whose value is at the range of $[0.25, 0.33]$

To make clustering to be suitable for data with higher coupling degree, we improve the Euclidean distance equation as following equation, to compute the price of exchanging center points:

$$d = (x_i - o_j)^T \times (x_i - o_j) \times DX_j \qquad \textbf{(1)}$$

d is exchanging price; x is data point, $i = 1, 2, ..., n$; o_j is representative object of cluster c_j, $j = 1, 2, ..., k$; DX_j is variance of cluster c_j. The introduction of DX can demonstrate the intimacy of cluster effectively: smaller variance means closer clusters and bigger density. The procedures of improved algorithm based on above descriptions are depicted as follows:

Input: Number of clustering k; Dataset including n data objects $Dataset$; Number of the maximum branches of non-leaf nodes $B = k$; Number of the maximum items of leaf node in CF tree L (The clustering accuracy between $B = L$ and $B \neq L$ makes little difference so $L = B$)

Output: Set of k clusters

```
Randomly select $N$ pairs of data object in $Dataset$;
Calculate the Euclidean distance $Ed$ of each pair of data;
Use $Ed$ to compute $EX$ and $DX$ ;
Set threshold $T$ as $P(EX + DX)$;
Establish CF tree according to $T$ and input branch factor $B$, to form micro-clusters;
Randomly select $k$ leaf nodes as initial object in the CF tree;
Repeat;
Assign surplus leaf nodes to the closest representative object $\{o_j\}$, $j = 1, 2, ..., k$ ;
Modify the CF value of non-leaf node recursively;
Recalculate the radius of all the leaf nodes. If it is larger than $T$, then begin node splitting (The rule is that selecting two data points with the farthest distance as new leaf nodes, and the other data points will be assigned into these two new nodes according to their distances); at the same time, delete original leaf node and update CF value of new non-leaf nodes.
Randomly select a non-representative object $o_r$ in the leaf nodes, $o_r \neq o_j$. Substitute the total price $S$ by $o_r$. If $S < 0$, then substitute $o_j$ by $o_r$ to form the set of $k$ new representative objects; else repeat doing until there is not change any more.
```

In this way we can acquire k clustering for improved CF tree. Since the micro-cluster compresses the data and reduces the number of data points, the algorithm can decrease the times of distance calculating. The variance of each clustering is taken into account, so our method can adapt to clustering with big density. Under many cases, it can decrease the time to update clustering center.

SVM training

After CF tree is established according to branch factor, it should be trained according to given threshold of local scale. We perform preorder traversal on the whole CF tree. For non-single-class sub-tree, we will directly train it to get local classifier of this sub-tree. The process in detail is described as the follows:

Input: A CF Tree T_{CF} , Local threshold T_h , SVN hyper- parameter σ

Output: CF tree Tr with local SVM classifier

```
(1)Traverse CF tree with preorder. When traversing a sub-tree, executing the following operations:
Check if this sub-tree is of single class, if so, record its label on the root node of the sub-tree and skip consequent traversing process; otherwise, go to procedure b;
Check if the scale of this sub-tree is larger that local threshold, if so, train this sub-tree directly and skip consequent traversing process; otherwise, continue consequent traversing process of the sub-tree.
(2)  Return CF tree with local SVM classifier.
```

During the training process, we need to make verifying tests for returned CF tree with local SVM classifier of each group of parameters, to find the optimal CF tree with SVM classifier. In the test, the samples are tested with optimal CF tree with SVM classifier and its specific process is described as follows:

Input: A test sample point, a CF tree with local SVM classifier

Output: Label of the sample point

```
Execute the following steps from root node;
Check if this node is a root node of single class sub-tree, if so, return recorded label of this node;
Check if this node is a root node of trained sub-set, if so, use its local classifier to classify this test samples node and return the acquired label;
Check if this node is leaf node, if so, search the CF item which has the closest distance to the test sample, and return the label of this CF item;
Search the CF item which has the closest distance to the node and execute steps 1-5 to corresponding children nodes.
```

EXPERIMENTAL RESULTS ANALYSIS

Large-scale dataset

In this experiment we use CFD-SVM to compare with some existing large-scale semi-supervised classification algorithms, including PVM$_{\text{square}}$, AGR, DTSVM, CVM, LibSVM and 1NN[22-24]. LibSVM and 1NN are taken as datum line. The dataset adopts mnist and extended dataset mnistex. TABLE 1 lists the learning accuracy of these algorithms on above dataset and TABLE 2 lists the learning time of each algorithm.

TABLE 1 : Learning accuracy of algorithms on large-scale dataset (%)

Algorithm	mnist	mnistex
LibSVM	92.08	87.01
1NN	88.34	83.22
PVM$_{\text{square}}$	93.09	89.08
AGR$_{\text{kernel}}$	94.32	90.11
AGR$_{\text{LAE}}$	93.85	91.01
CVM	93.33	92.19
DTSVM	93.98	93.07
CFD-SVM	95.10	93.80

TABLE 2 : Learning time of algorithms on large-scale dataset (s)

Algorithm	mnist	mnistex
LibSVM	133.25	225.13
1NN	4.51	43.18
PVM_{square}	71.82	850.65
AGR_{kernel}	68.90	2494.79
AGR_{LAE}	188.26	3876.23
CVM	91.22	725.33
DTSVM	93.10	770.18
CFD-SVM	93.08	625.76

Above results show that CFD-SVM has better accuracy among the algorithms. On mnist dataset, the learning time of CFD-SVM is smaller than PVM and AGR and close to LibSVM. PVM and AGR are global algorithms. Though they can reduce the scale of inverse matrix by selecting prototype vector and anchor similar constructing matrix for label reconstruction, the scale of other matrix operations such as product of similar matrix of all the samples with prototype vector and anchor is still large. When data scale is very big, the influence on learning speed of large matrix will stand out. So the learning time on mnistex is higher than others, CFD-SVM included. In addition, the demand for memory of PVM and AGR is also higher than our algorithm. With the increase of data scale, they tend to reduce the data of prototype vector and anchor to be suitable for the following work. But they are based on above factors for label broadcasting. So the reduction of label points will decrease their accuracy, which will not occur in CFD-SVM. Because CF tree can be established dynamically according to limited memory. Secondly, the complexity to choose key points by k-means clustering and the scale of local learning are controllable. So CFD-SVM keeps good applicability without reducing the number key points.

To obviously describe the improvement on accuracy of our algorithm, compared to classical supervised algorithms, Figure 4 offers the improvement of accuracy in percents. It demonstrates that, most semi-supervised algorithms acquire substantial upgrade in accuracy compared to 1NN; For SVM, except DTSVM and CFDSVM, increasing 3% averagely, the others do not show advantage of improvement.

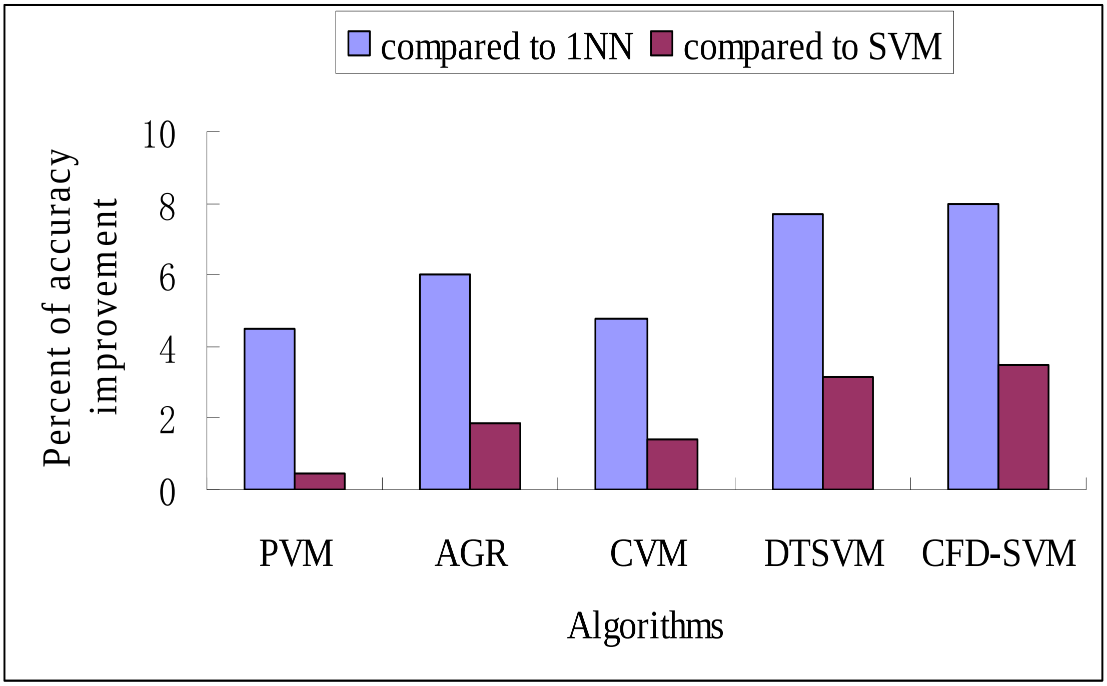

Figure 4: The improvement on accuracy of algorithms compared to 1NN and SVM.

Ultra-large-scale dataset

To measure the influence of memory to CFD-SVM, we add the memory from 0.1GB to 0.5 GB, 0.1GB per time. TABLE 3 shows the available memory on our algorithm. The size of available memory has obvious influence on compression proportion and test accuracy and larger memory can obtain bigger proportion and test accuracy. It demonstrates that our algorithm will automatically adjust the compression degree of data according to the memory size, which inherits the advantage of BIRCH, to finish training of large-scale data with limited memory.

TABLE 3 : Compression proportion and test accuracy of CFD-SVM with different size of memory

Indicator	Dataset	0.1GB	0.2GB	0.3GB	0.4GB	0.5GB
compression	covtype	17.89	24.33	27.36	28.99	100
	forest	17.96	25.51	25.79	28.23	100
Test accuracy	covtype	90.44	92.33	91.95	93.42	94.86
	forest	89.28	90.93	90.84	91.27	92.40

There occurs memory overflow in the test of KDD-full for DTSVM, so its experimental result can not be acquired. While the dataset of “KDD-full” is sparse, we call load the memory with sparse forms. Then the results of CVM and LibSVM can be obtained. For the experiment of “minst8m”, the memories of LibSVM, CVM and DTSVM are all insufficient during test process. So we do not load this dataset.

TABLE 4 lists some optimal parameters on the ultra-large-scale datasets for CFD-SVM, CVM and LibSVM. The experiments results which can not be acquired are replaced by ϕ in corresponding blanks in the table. TABLE 5 lists the training time, test accuracy and test time of above three algorithms on the ultra-large-scale datasets.

TABLE 4 : Optimal parameters of CFD-SVM, CVM and LibSVM in two datasets

Algorithm	CFD-SVM			LibSVM		CVM	
Dataset and parameters	η	C	γ	C	γ	C	γ
KDD-full	12.17	10000	1	10	1	1000	0.1
Mnist8m	13.51	10	1	ϕ	ϕ	ϕ	ϕ

TABLE 5 : Compression proportion and test accuracy of CFD-SVM with different size of memory

Dataset	Algorithms	Training time (s)	Test accuracy (%)	Test time (s)
KDD-full	CFD-SVM	150.203	99.98	14.56
	CVM	199.215	99.91	2944.206
	LibSVM	4900.827	99.98	729.881
Mnist8m	CFD-SVM	531.274	98.67	238.15

The experimental results on ultra-large-scale datasets are analyzed as the follows: First, for training time and test time, CFD-SVM is fast that CVM and LibSVM on sparse dataset KDD-full. In the test speed, it is almost the same with the other algorithms; Second, in mnist8m, when the other three algorithms can not finish the computation due to adequate memory, our algorithm still works and shows higher efficiency.

CONCLUSION

In the field of machine learning, the classification of large-scale data has been the emphasis of study. SVM algorithm, as a well efficient classification method, is wildly applied to every fields of machine learning. With the future development of large-scale data age, large-scale SVM algorithms will make more contributions in machine learning and data mining. By the research on SVM and hierarchical clustering algorithms, this paper proposes a large-scale classification algorithm with CF tree and SVM, based on the idea of local learning. It adopts CF tree to separate samples to make local learning, which reduces the scale of study. Then by improved distance measuring method, it constructs more accurate local classifiers by CF tree. These local classifiers are integrated into the global classifiers at last to test and train the dataset. Experiments show CFD-SVM inherits some advantages of hierarchical clustering algorithm BIRCH, such as scalability, data compression, memory utilization and low times of scanning. It also keeps the advantage of high test accuracy of SVM algorithms. Compared to traditional classifiers, our scheme has better comprehensive performance.

Though CFD-SVM acquires relatively better results in scale-data classification, there still exists some factors to be improved, including:

Main of CFD-SVM is based on BIRCH, so it has low scalability on high-dimensional dataset. We should study the hierarchical structure of the algorithm under such environments, to overcome the defect on dimensions;

The idea to separate dataset with clustering methods is essentially a data reduction technology, which lacks some theoretical proofs. So the next work is quantifying the relation between local scale threshold and the test accuracy, to make it more convincing.

REFERENCES

[1] Kurc Tahsin, Uysal Mustafa, Eom Hyeonsang; Efficient performance prediction for large-scale, data-intensive applications, International Journal of High Performance Computing Applications, **14(3)**, 216-227 **(2000)**.

[2] Radi Mohammed, Mamat Ali, Deris M.Mat; Framework for evaluating update propagation techniques in large scale data grid, Proceedings of 1st International Conference on Distributed Frameworks and Application, Penang, Malaysia, 89-95 **(2008)**.

[3] Wang Xili, Liu Fang, Jiao Licheng; Training and classification of SVMs based on large-scale data, Jounal of Xidian University, **29(1)**, 123-127 **(2008)**.

[4] Eliassi-Rad Tina, Critchlow Terence, Abdulla Ghaleb; Statistical modeling of large-scale simulation data, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Canada, 488-494 **(2002)**.

[5] Wang Lijun, Dong Ming; On the clustering of large-scale data: A matrix-based approach, Proceedings of the International Joint Conference on Neural Networks, San Jose, USA, 139-144 **(2011)**.

[6] R.C.Chen, C.H.Hsieh; Web page classification based on a support vector machine using a weighted vote schema, Expert Systems with Applications, **31(2)**, 427-435 **(2006)**.

[7] H.P.Graf, E.Cosatto, L.Bottou, et al; Parallel support vector machines, Advances in Neural Information Processing Systems, **17**, 521-528 **(2005)**.

[8] H.Zhang, A.C.Berg, M.Maire, et al; SVM-KNN: discriminative nearest neighbor for visual object recognition, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2126-2136 **(2006)**.

[9] F.Chang, C.Y.Guo, X.R.Lin, et al; Tree Decomposition for Large-Scale SVM Problems, Journal of Machine Learning Research, **11(1)**, 2935-2972 **(2010)**.

[10] Zhang Yanfang, Li Jinhong, Cao Danyan; Improvement of the K-means Clustering Algorithm Based On CF-tree, Software Guide, **15(2)**, 42-45 **(2005)**.

[11] Huang TianQiang, Qin XiaoLin, Wang JinDong; Multi representation Feature Tree and Spatial Clustering Algorithm, Computer Science, **33(12)**, 189-194 **(2006)**.

[12] S.S.Keerthi, S.K.Shevade, C.Bhattacharyya, et al; A fast iterative nearest point algorithm for support vector machine classifier design, IEEE Transactions on Neural Networks, **11(1)**,124–136 **(2000)**.

[13] Li Wenjuan, Hu Chunsheng; A Combined Learning Algorithm of Optimum Covering Based on Clustering Computer Technology and Development, **20(11)**, 51-58 **(2010)**.

[14] H.Yu, J.Han, J.Yang, et al; Making SVMs scalable to large data sets using hierarchicalcluster indexing, Data Mining and Knowledge Discovery, **11(3)**, 295–321 **(2005)**.

[15] Zhou Yingchun, Luo Jiawei; An improved BIRCH Clustering Algorithm and its applications, Journal of Zhanjiang Normal College, **30(3)**, 83-87 **(2010)**.

[16] Wei Xiang; Improved BIRCH clustering algorithm based on density, Computer Engineering and Applications, **49(10)**, 201-205 **(2013)**.

[17] Tao Jianwen, Wang Shitong; Local learning based support vector machine, Control and Decision, **27(10)**, 1510-1515 **(2012)**.

[18] J.Su, J.Sayyad-Shirabad, S.Matwin; Large Scale Text Classification using Semi-supervised Multinomial Naive Bayes, Proceedings of 28th International Conference on Machine Learning, Bellevue, WA, USA, **(2011)**.

[19] Lei Xiaofeng, Yang Yang, Zhang Ke; Metaheuristic Strategy Based K-Means with the Iterative Self-Learning Framework, Computer Science, **36(7)**, 175-178 **(2009)**.

[20] Ren Peihua; Data Flow Fast Clustering Algorithm Based on Micro Cluster Evolution Learning, Computer Simulation, **30(3)**, 343-347 **(2013)**.

[21] Shixue Zhang, Jinyu Zhao; Feature Aware Multiresolution Animation Models Generation, Journal of Multimedia, **5(6)**, 322-628 **(2010)**.

[22] Feng Yining, Shao Yuanhai, Chen Jing; Hierarchical clustering based on weighted SVM in large-scale data, Computer Engineering and Design, **30(1)**, 175-178 **(2006)**.

[23] K.Nigam, A.McCallum, S.Thrun, et al; Text Classification from Labeled and Unlabeled Documents using EM, Machine Learning, **39(2)**, 103-134 **(2000)**.

[24] Shen Yan, Song ShunLin, Zhu YuQuan; A clustering algorithm for scalable datasets based on semi-supervision technology, Journal of Nanjing University (Natural Sciences), **47(4)**, 373-382 **(2011)**.