8-2008

Classification in P2P Networks by Bagging Cascade RSVMs

Hock Hee ANG
Nanyang Technological University

Vikvekanand GOPALKRISHNAN
Nanyang Technological University

Steven C. H. HOI
Singapore Management University, CHHOI@smu.edu.sg

Wee Keong NG
Nanyang Technological University

Anwitaman DATTA
Nanyang Technological University

Classification in P2P Networks by Bagging Cascade RSVMs

Hock Hee Ang, Vivekanand Gopalkrishnan, Steven C.H. Hoi, Wee Keong Ng, and Anwitaman Datta

Nanyang Technological University
{angh0024,asvivek,chhoi,awkng,anwitaman}@ntu.edu.sg
http://www.ntu.edu.sg

Abstract. Data mining tasks in P2P are bound by issues like scalability, peer dynamism, asynchronism, and data privacy preservation. These challenges pose difficulties for deploying conventional machine learning techniques in P2P networks, which may be hard to achieve classification accuracies comparable to regular centralized solutions. We recently investigated the classification problem in P2P networks and proposed a novel P2P classification approach by cascading Reduced Support Vector Machines (RSVM). Although promising results were obtained, the existing solution has some drawback of redundancy in both communication and computation. In this paper, we present a new approach to over the limitation of the previous approach. The new method can effectively reduce the redundancy and thus significantly improve the efficiency of communication and computation, meanwhile it still maintains good classification accuracies comparable to both the centralized solution and the previously proposed P2P solution. Experimental results demonstrate the feasibility and effectiveness of the new P2P classification solution.

1 Introduction

Data mining in Peer-to-Peer (P2P) networks has recently attracted considerable interests due to abundant knowledge within the data that is distributed over a large number of peers in the networks. For instance, performing clustering and classification on network traffic or stored files may reveal behaviors and relationships among peers. In addition, some typical large-scale data mining tasks that are computationally very intensive with a traditional centralized approach would become feasible and practical with a P2P solution.

Ideally, a data mining technique in P2P networks, e.g., P2P classification or P2P clustering, is expected to achieve learning performance that is comparable to that of a regular centralized approach. This, however, is a very difficult task. For instance, P2P classification (also P2P clustering), often faces a number of challenges [1], such as *scalability* (Can the algorithm produce good results within an acceptable duration when there are many peers?), *peer dynamism* (Can the algorithm handle the availability and unavailability of data as peers connect and disconnect from the network?), *asynchronism* (Can the algorithm provide

sufficiently accurate results without global synchronization?), and *data privacy* (Can the algorithm protect the privacy of peers' data when learning in the global environment?).

This paper studies the research problem of P2P classification by exploring cascade learning techniques [2, 3, 4, 5] in the context of learning in P2P networks. In particular, we are interested in improving the efficiency of cascade learning for classification in a P2P network, while maintaining accuracy comparable to that of a centralized approach. Recently, we presented Cascade Reduced Support Vector Machines (RSVM) for P2P networks, which classifies in P2P networks by cascading local models of peers in order to create the global model [6]. Although this approach delivers promising results, it relies on excessive model propagation and model computation by peers, and results in possibly long waiting times before satisfactory classification accuracy can be achieved.

In this paper, we propose a new approach for P2P classification by combining cascade learning with the concept of bootstrap aggregation (bagging) [7, 8, 9] for learning classifiers in P2P networks. The new solution reduces the redundancy of our previous approach, and thus significantly reduces the communication and computation cost while maintaining accuracy comparable to that of centralized solutions. The feasibility and effectiveness of our approach are validated by extensive experimental evaluations.

As a summary, our contributions in this paper are as follows: (1) We investigate the problem of P2P classification in P2P network and address limitations of our recently proposed solution with P2P Cascade RSVM; (2) We propose a new approach of bagging Cascade RSVMs for P2P classification, which significantly reduces the costs of both training computation and data communication; (3) We conduct a set of extensive experiments for comparing the efficiency and accuracy of the proposed new solution with existing approaches, in which promising results validate the effectiveness of our technique.

The rest of this paper is organized as follows. Section 2 reviews the background and related work. Section 3 presents our proposed approaches for performing classification learning in P2P networks. Section 4 presents our experimental results and Section 5 concludes this paper.

2 Related Work

2.1 Background Overview

Let $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ denote a set of training data, where $\mathbf{x}_i \in \mathbb{R}^d$ is a d-dimensional data point and $y_i \in \mathcal{Y}$ is the corresponding class label, and $\mathcal{Y}$ is the class label space, e.g. $\{+1, -1\}$ for a binary classification task. The goal of a classification task is to learn an effective classifier from the data $f : \mathbb{R}^d \rightarrow \mathcal{Y}$, such that class labels of unseen data can be predicted correctly.

Many popular classification techniques, such as decision tree, nearest neighbor classifiers, artificial neural networks, Bayes classifiers and support vector machines (SVM), can perform very well on small-sized datasets. However, they usually cannot scale well on very large data in real world scenarios in terms

of heavy time and memory costs. Hence, alternative solutions, such as selective sampling [10, 11, 12], and parallelized and distributed learning [13, 14, 2, 3, 4, 5], have been investigated for scalable learning from large datasets.

Pasting of Ivotes was introduced by Breiman [10] to train an ensemble of classifiers using importance sampling based on out-of-bag estimation. Other ensemble approaches aimed at improving accuracy include Bagging [7, 8, 9], which performs plurality voting on an ensemble of classifiers built with bootstrapped datasets and Boosting [15], which creates a series of classifiers each of them focuses more on the previously wrongly classified instances.

Lee and Mangasarian [11] presented a different approach, the Reduced Support Vector Machines (RSVM), which uses a randomly chosen smaller subset of data to solve the SVM optimization problem. To provide further understanding of the RSVM, Lin and Lin [12] studied several implementations of RSVM and illustrated that RSVM can significantly reduce training time with a slight decrease in accuracy compared to SVM for problems with dense support vectors.

Parallelized and distributed algorithms present alternative approaches to solve very large scale classification problems, and can be broadly categorized as ensemble and cascade approaches. Typically, these algorithms employ the divide-and-conquer paradigm to split a large problem into smaller sub-problems each of them is much easier to be solved, and finally merge their results to produce the final global solution. One positive side effect of these algorithms is that they can also be applied on *naturally distributed* data, whereas learning with a centralized approach would require moving the data to a single location. Such approaches include DIvotes [16], a distributed version of Ivotes, and distributed boosting [14], which propagates local training statistics to other sites in order to perform adaptive learning. Unlike DIvotes, which requires no communication between peers during training, distributed boosting's propagation of training statistics during training can significantly increase the communication overhead.

In order to address the scalability issue of SVM, Tveit and Engum [2] presented a heap-based tree topology framework for parallelizing the computation of Proximal SVM , which is the pioneering work on Cascade SVM. Since then, many more research efforts have concentrated on Cascade SVM [3, 4, 5]. For instance, Lu *et al.* [3] presented and compared several ways of cascading SVM, Zhang *et al.* [4] examined various ways to incorporate feedback for improving accuracy of cascade SVM and Graf *et al.* [5] presented their cascade SVM approach and proved the convergence of their algorithm.

2.2 Learning in P2P networks

In recent years, with the increasing popularity of P2P networks, classification problems in P2P networks have also gained a lot of attention. A P2P network $P = \{p_1, \ldots, p_N\}$ is a set of N interconnected heterogeneous peers, where all peers perform the same functions with no definition of client or server. Learning in P2P networks can be very challenging due to the characteristics of the P2P networks such as scalability, peer dynamism, data dynamism, asynchronism, privacy and security [1]. In a P2P network, the number of peers N usually exceeds

hundreds or thousands characterizing it as a massively distributed system. To further complicate matters, peers in a P2P network may leave or join the network at anytime and the data that these peers hold change frequently. Given the size of the network, synchronization of the network is almost infeasible considering the network latency and bandwidth. Furthermore, privacy and security will become a concern if peers were to exchange data.

Depending on how the data is propagated, existing P2P classification techniques can be classified as model propagation [17] or test data propagation [18, 19] approaches. Model propagation techniques propagate the peers' model that is built on local data to other peers and these collected models are used to generate the final hypothesis using methods such as plurality voting or meta-learning. For instance, Siersdorfer and Sizov [17] proposed to propagate SVM models for classifying web documents in a P2P network. The choice of SVM is to generate compact models to reduce the communication overhead, but for some dense problems, SVM still generates a lot of support vectors. In addition, they proposed to tune the global model through synchronization to improve accuracy, at the expense of incurring more communication cost. Generally, model propagation approaches incur high communication cost during model construction which aggravates if the local model changes frequently. However, the advantage of these approaches is that prediction does not require communication among peers, reducing the communication cost and peers have the option to freely manipulate the collected models.

Test data propagation approaches, on the other hand, only propagate test data to other peers, which in turn reply with the classification results obtained using their local models. Although test data propagation approaches do not incur communication cost during model construction, given frequent prediction tasks, their communication cost may exceed that of model construction. Recently, Luo *et al.* [19] presented a P2P version of the Ivotes where each peer pastes small bites to build a local classifier until the out-of-bag error rate falls below a specified threshold. Classification is then performed by using an optimal communication protocol to propagate the test instances to gather votes of the peers in the P2P network.

3 Approach

We now present the design details of our approach, which is a generalization of our recently proposed P2P Cascade RSVM [6], and tries to improve communication and computational efficiency. As shown in the next section, by the new approach, communication and computation costs can be reduced by up to 40 and 60 percent respectively, while comparable accuracy is maintained. P2P Cascade RSVM mainly consists of three stages, first local SVMs are computed, next the models are propagated, and finally the collected models are merged.

3.1 P2P Cascade RSVM

In the first stage, each peer builds a model using RSVM on the local data. The resulting model is a very compact representation of the local data since only a

small subset of the local data is used in solving the optimization problem (support vectors are chosen from the subset). The use of RSVM not only reduces the communication overhead in model propagation by generating a very compressed model, but also improves the computational efficiency by working on a smaller subset of data.

After the local models are constructed, they are propagated to other peers. Model propagation reduces the effect of peer dynamism as the propagated model exists in the network even after the creating peer goes offline (assuming the local model was propagated before peer goes offline). Although model propagation places a large burden on the communication overhead, as stated above, this effect is significantly reduced through the use of RSVM.

Once a peer receives other peers' models, it merges them to build a Cascade SVM. The cascaded SVM is a representative of all the merged models, hence classification can be performed by using only the cascaded SVM.

Although P2P Cascade RSVM uses RSVM to create a very compact representation of the local data, the collection of all models in the entire P2P network still results in a high communication overhead. Moreover, it may take quite a long time before a substantial number of models are collected, and this definitely affects classification accuracy in the initial stages.

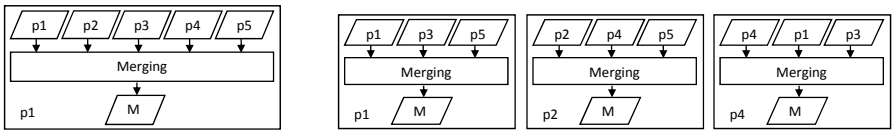
To overcome the existing limitations present in the plain use of Cascade RSVM in P2P networks, we propose a variant of the P2P Cascade RSVM, named P2P Bagging Cascade RSVM, which integrates the concept of bootstrap aggregating (*bagging*) in order to improve the communication and computation efficiency. The main differences with P2P Cascade RSVM lie in the model propagation and the prediction process.

3.2 P2P Bagging Cascade RSVM

Similar to P2P Cascade RSVM, each peer creates a local model from local data using RSVM. However, instead of receiving models from all other peers, a peer now only collects models from k randomly chosen peers and merges the collected models to build the local cascade SVM. Note that when k is equal to the number of peers in the entire network, P2P Bagging Cascade RSVM reduces to the P2P Cascade RSVM. Having many different peers building different cascade SVMs from these randomly chosen models, closely resembles the creation of an ensemble of classifiers based on random sub-sampling. Since only a subset of the peer's models are worked upon by a peer, it incurs significantly lesser communication (collecting models) and computation (merging models) costs. Moreover, the collection and merging of models are performed in parallel by peers. Also, by requiring less peers' participation, our new approach is more fault tolerant.

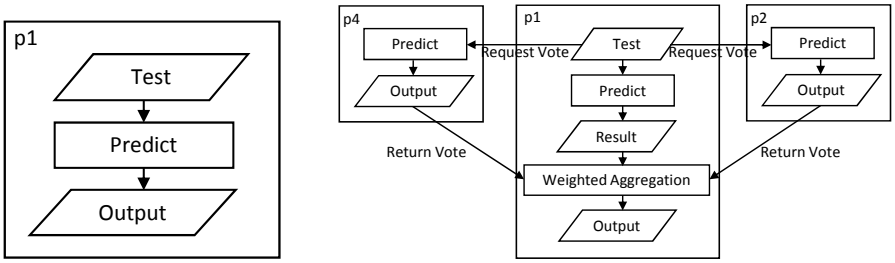
Finally, let us now look at how prediction is carried out in P2P Bagging Cascade RSVM. Contrary to P2P Cascade RSVM, which predicts with only the local cascaded SVM, P2P Bagging Cascade RSVM predicts based on weighted majority voting of numerous peers' cascaded SVMs. After the construction of the cascaded SVMs, each peer evaluates performance of the built model using the

training data, and uses that value as the weight of its prediction. Test instances are propagated to $v-1$ randomly chosen peers for voting, where the predicted class is returned together with the peer's weight. The final prediction is then obtained by aggregating the weights (total of v votes inclusive of the local model) and choosing the class with the greatest weight. Although involving all peers in the voting process may produce the optimal result, we empirically show that only a small number of votes is required to achieve satisfactory classification accuracy, after which additional time and communication costs do not justify the small increase in accuracy.



(a) P2P Cascade RSVM (b) P2P Bagging Cascade RSVM

Fig. 1. Example of training phase for the two P2P Cascade RSVM approaches: total number of peers (N) is 5, models to be cascaded (k) for P2P Bagging Cascade RSVM is 3 and number of peers to vote (v) is 3



(a) P2P Cascade RSVM (b) P2P Bagging Cascade RSVM

Fig. 2. Example of prediction phase for the two P2P Cascade RSVM approaches: total number of peers (N) is 5, models to be cascaded (k) for P2P Bagging Cascade RSVM is 3 and number of peers to vote (v) is 3

To further clarify the differences between the P2P Cascade RSVM and P2P Bagging Cascade RSVM, Figure 1 and 2 give an example to illustrate the training and prediction phases, respectively. In this example, the total number of peers in the network (N) is 5, the number of models to be collected(k) is 3 and the number of voting peers (v) is 3. The data (parallelogram) labeled p_i and M represent the collected local models of peer i and the local cascaded model respectively. The outer bounding box represents the locality of the training process. As shown, during the training phase, P2P Cascade RSVM collects all models in the network, whereas P2P Bagging Cascade RSVM only collects k randomly selected models (inclusive of its own model) for merging. During prediction, P2P Cascade RSVM only requires the local cascaded model, however, P2P Bagging Cascade RSVM requests for the votes of $v-1$ peers, and performs weighted aggregation. Finally, Algorithm 1 shows the details of model construction in each peer. For limited space, we skip the algorithm for the prediction phase.

Algorithm 1: Model Construction in peer p_i

input: percentage of support vectors to use (p), number of models to collect/cascade (k), local training data (D_i)

$SSV_i = \{\}$; $PSV_i = \{\}$; $models\_collected = 0$; training data $T = \emptyset$
Train local classifier model M_i using RSVM on D_i
$models\_collected = 1$
while $models\_collected < k$ **do**
    Collect the support vectors SV_j of a randomly chosen peer j
    if $SV_j \notin SSV_i$ *and* $SV_j \notin PSV_i$ **then**
        $PSV_i = PSV_i \cup SV_j$
    if PSV_i *is not empty* **then**
        $T =$ support vectors of M_i
        forall $SV \in PSV_i$ **do**
            $T = T \cup SV$
        $M_i =$ SVM model trained using T
        $SSV_i = SSV_i \cup PSV_i$
        $PSV_i = \{\}$

3.3 Cost Analysis

Here we present a simple summary of computation and communication costs of the various algorithms, including the centralized RSVM, SVM Ensemble, P2P Cascade RSVM, and P2P Bagging Cascade RSVM. Let l be the total size of the problem, N be the total number of peers and k and v be the number of models to collect/cascade and the number of peers to vote respectively in P2P Bagging Cascade RSVM. Given s as the percentage of the data to use for RSVM, we let $m = ls, m << l$ be the size of the data subset.

Table 1. Summary of the training cost.

Approach	Computation Cost	Communication Cost
SVM	$O(l^3)$	$O(l)$
RSVM	$O(lm^2)$	$O(l)$
P2P Cascade RSVM	$O(m^3)$	$O(m)$
P2P Bagging Cascade RSVM	$O((km/N)^3)$	$O(km/N)$

The summary of the computation and communication cost for training is given in Table 1. We note that in [6], the computation complexity is given as $O(m^3)$ with the merging cost based on that of traditional SVM although it can be reduced with the use of SMO algorithms (still greater than $O(l^2)$). P2P Bagging Cascade RSVM, by using only a subset of the peers' models, $k < N$, is able to provide a quadratic reduction in computation ($O((km/N)^3)$) and a linear reduction in communication cost $O(km/N)$. From Table 1, we can see that P2P Bagging Cascade RSVM's computation and communication is significantly less than the other approaches.

4 Experimental Results

In this section, we present the results of our empirical experiments for evaluating the performance of our technique compared with several competing approaches. In our experiments, we adopt some large data sets in order to mimic possible real-world P2P scenes. For the rest of experiments, we first give an overview of the experiments setup and then compare the classification accuracies of various classification algorithms, including both centralized and P2P approaches. We further examine the effects of parameter selection with respect to the resulting classification accuracy. Finally we provide some cost analysis based on time and communication.

4.1 Experimental Setup

In the experiments, we have used the covertype dataset from the UCI repository [20] and the MNIST dataset [21]. The Covertype dataset, is a collection of forest cover type data from the UCI repository, which is one of the largest datasets available. The MNIST dataset is a handwriting digit database that is widely used for pattern recognition benchmarks. The original Covertype dataset was used to generate another dataset, the binary Covertype, which contains only two classes (class two versus all other classes). For all experiments, 10-fold cross validation is performed on both binary covertype and covertype dataset, while the MNIST dataset is trained and tested with the regularly partitioned training and test sets. All attributes of the datasets were normalized to the range of [-1,1]. Table 2 gives a summary of the datasets in our experiments.

Table 2. Summary of the datasets used in experiments.

	Instances	Attributes	Classes
Binary Covertype	581,012	54	2
Covertype	581,012	54	7
Mnist	70,000	21	10

For the experimental environment, we have employed a cluster of 16 machines, each of them has two Intel Dual Core Xeon 3.0GHz processors, 4-GB RAM and is connected by a gigabit ethernet to conduct the experiments.

Due to the large size of the training data, we have adopted the Reduced SVM (RSVM) algorithm as the baseline centralized SVM solution, which is implemented in C++ and available from LIBSVM [12]. In addition, the same RSVM code is used for building the initial local models in our approach, and the C-SVM [12, 22] is used when building the cascading models.

In the experiment, 500 peers were simulated for both the binary Covertype and multi-class Covertype datasets, and 100 peers were used for the MNIST dataset. For the binary Covertype and multi-class Covertype datasets, one percent of the data was used for building the initial RSVM model, and five percent was used for the MNIST dataset. The selection of the specific percentages is to ensure that each peer has sufficiently data for training. For parameter selections, the RBF kernel was used all RSVM and SVM based algorithms, and the γ and C values were selected by the model selection tool provided with LIBSVM based on a one percent of stratified sampled data from each of the datasets.

4.2 Classification Accuracy

We evaluate the classification performance of our proposed P2P Bagging Cascade RSVM approach with several competing solutions, including the baseline centralized RSVM, the SVM Ensemble that is a combination of multiple SVMs, and the recently proposed P2P Cascade RSVM [6]. For the setting of the P2P Bagging Cascade RSVM approach, we engaged 50 percent of the peers' models for cascading and 10 percent of the peers for voting. Hence, for the binary Covertype and multi-class Covertype datasets, the number of models cascaded $k = 250$, and the number of peers for voting $v = 50$, and for the MNIST dataset, $k = 50$ and $v = 10$.

Table 3 shows the experimental results of both classification accuracies and training cost on the three datasets. Several observations can be drawn from the results. First of all, we found that both proposed P2P solutions are able to achieve good classification accuracy performance comparable to the centralized baseline. Secondly, in term of training cost, both P2P solutions are significantly more efficient than the centralized baseline of RSVM. Finally, we found that the newly proposed solution, P2P Bagging Cascade RSVM, is more efficient than the previous P2P Cascade RSVM approach with comparable classification accuracies. These promising results validate the effectiveness of our new technique.

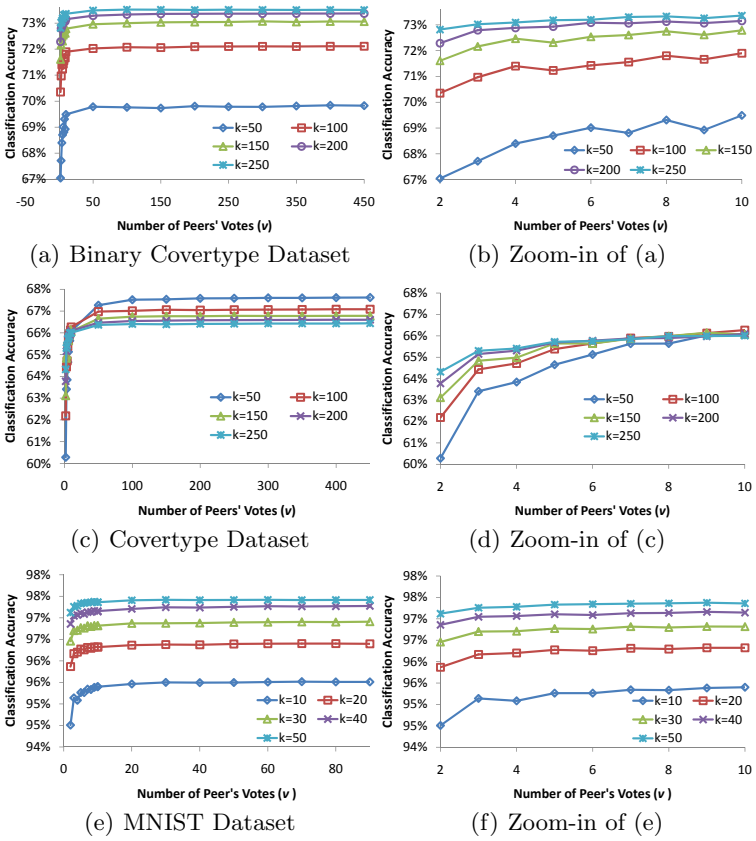
Table 3. Experimental results on three datasets.

Dataset	Accuracy (Training Time)			
	RSVM	SVM Ensemble	P2P Cascade RSVM	P2P Bagging Cascade RSVM
Binary Covertype	71.97%(111.2s)	52.35%(47.9s)	72.93%(5.6s)	**72.99%**(1.5s*)
Covertype	**67.16%**(695.7s)	46.41%(37.9s)	65.6%(151.6s)	66.37%(17.8s*)
MNIST	96.97%(1082.0s)	93.40%(56.0s*)	**97.85%**(189.0s)	97.36%(69.9s)

4.3 Effects of Parameter Selection

To further examine how will the parameters affect the performance of the proposed P2P Bagging Cascade RSVM solution, we conduct a set of experiments for evaluating the performance of different parameters used for the number of peers' models cascaded k and the number of peers involved for voting v. In particular, we perform the experiment by varying the value of k from 10 to 90 percent of all peers and the value of v from from 2 to 90 percent of all peers. Figure 3 shows the experimental results by fixing one parameter (k) and varying the other parameter (v) on the three datasets.

Some observations can be drawn from the results. First of all, on both the binary Covertype and MNIST datasets, we can clearly see the positive correlations between the accuracy and the number of peers voted (v) or the number of models cascaded (k) when the values of k and v are small. But the accuracy performance starts to reach a plateau when k exceeds 40 percent of all models (200 in Covertype and 40 in MNIST) and v exceeds 10 percent of all votes (50 in Covertype and 10 in MNIST). The results on the multi-class Covertype dataset, on the other hand, exhibited some different behaviors, as shown in Figure 3(c). The similar behavior of positive correlation can still be observed for the cases



(a) Binary Covertype Dataset (b) Zoom-in of (a)

(c) Covertype Dataset (d) Zoom-in of (c)

(e) MNIST Dataset (f) Zoom-in of (e)

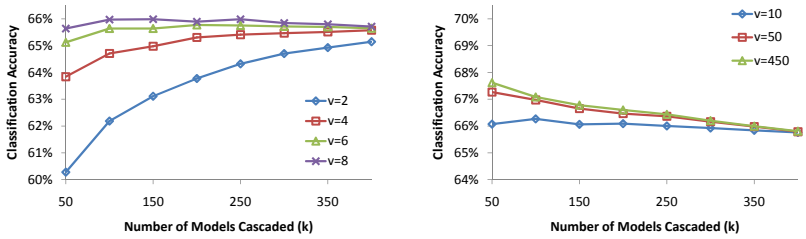
Fig. 3. Effects of parameter selection (k and v) on accuracy for P2P Bagging Cascade RSVM: Each line in the plots represents a fixed number of models cascaded (k) with varying values of peers voting (v).

with small v values as shown in Figure 3(d). But for the cases with large v values, the increase of k actually causes the the drop of accuracy.

To examine the above issue in more details, we further evaluate the situations by fixing the parameter v and varying the parameter k as shown in Figure 4. The results are similar to the previous observations. One possible reason is that the increase of k cause the drop in the variance of the ensembles of cascaded classifiers, and thus affects the classification performance as bagging works best when the variance in classifiers are maximized. In fact, when k reaches the maximum value, the P2P bagging cascade RSVM solution reduces to the previous P2P cascade RSVM approach. This result again validate the effectiveness and significance of the proposed new techniques.

4.4 Cost Evaluation

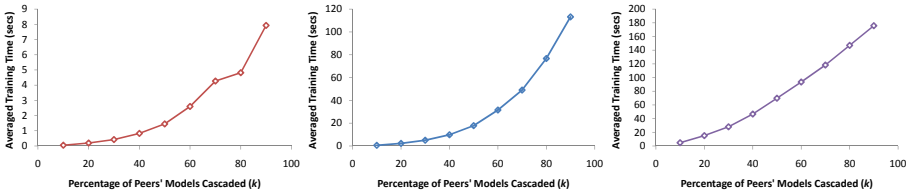
The last experiment is to evaluate the computational cost of the proposed solution. Figure 5 shows the average time taken for training the P2P Bagging



(a) $v < 10$ (b) $v \geq 10$

Fig. 4. Effects of number of peers voting (v) on accuracy for covertype dataset.

Cascade RSVM with each peer. The P2P Cascade RSVM, as a special case of the P2P Bagging Cascade RSVM with $k = N$ and $v = 0$, lies in the extreme right ends of the figures, which are the most costly cases for model construction. From the results on all datasets, we observed that the average time taken for training the P2P Bagging Cascade RSVM increases quadratically with respect to the number of models cascaded. Hence, a linear reduction in the number of models cascaded will provide a quadratic reduction in computation cost of the previous P2P Cascade RSVM solution. In addition to the advantage of training efficiency, the proposed P2P Bagging Cascade RSVM approach enjoys a smaller communication cost for model collection and voting, which increases linearly with respect to the number of models and the number of voters, respectively. In another words, a linear reduction in either the number of models cascaded or the number of voters involved will yield a linear reduction of the communication cost compared with the previously P2P Cascade RSVM approach.



(a) Binary Covertype Dataset (b) Covertype Dataset (c) MNIST Dataset

Fig. 5. Effects of parameter selection (k and v) on accuracy for P2P Bagging Cascade RSVM: Each line in the plots represents a fixed number of models cascaded (k) with varying values of peers voting (v).

5 Conclusion

We studied the classification problem in P2P networks and proposed a new approach for extending our recent work on P2P Cascade RSVM by incorporating the concept of bagging, which aims at improving the efficiency. We conducted extensive experiments for evaluating both efficiency and accuracy performance. Experimental results showed that our new approach is able to substantially reduce computation and communication cost, meanwhile achieve accuracy comparable to our previous approach and the centralized solution. In future work,

we will study more effective voting schemes for taking into consideration the distribution of data, data privacy and security as well as fault tolerance.

References

[1] Datta, S., Bhaduri, K., Giannella, C., Wolff, R., Kargupta, H.: Distributed data mining in peer-to-peer networks. IEEE Internet Computing, Special issue on Distributed Data Mining **10**(4) (2006) 18–26

[2] Tveit, A., Engum, H.: Parallelization of the incremental proximal support vector machine classifier using a heap-based tree topology. IDI, NTNU, Trondheim, (2003)

[3] Lu, B., Wang, K., Wen, Y.: Comparison of parallel and cascade methods for training support vector machines on large-scale problems. Machine Learning and Cybernetics **5** (2004) 3056–3061

[4] Zhang, J., Li, Z., Yang, J.: A parallel SVM training algorithm on large-scale classification problems. Machine Learning and Cybernetics **3** (2005) 1637–1641

[5] Graf, H.P., Cosatto, E., Bottou, L., Dourdanovic, I., Vapnik, V.: Parallel support vector machines: The cascade svm. Advances in Neural Information Processing Systems **17** (2005) 521–528

[6] Ang, H.H., Gopalkrishnan, V., Hoi, C.H., Ng, W.K.: Cascade rsvm in peer-to-peer network. Principles of Data Mining and Knowledge Discovery (2008) (To appear)

[7] Breiman, L.: Bagging predictors. Machine Learning **24**(2) (1996) 123–140

[8] Geurts, P.: Some enhancements of decision tree bagging. Principles of Data Mining and Knowledge Discovery (2000) 136–147

[9] Kim, H.C., Pang, S., Je, H.M., Kim, D., Bang, S.Y.: Support vector machine ensemble with bagging. SVM (2002) 397–407

[10] Breiman, L.: Pasting small votes for classification in large databases and on-line. Machine Learning **36**(1-2) (1999) 85–103

[11] Lee, Y., Mangasarian, O.: Rsvm: Reduced support vector machines. First SIAM International Conference on Data Mining (2001) 00–07

[12] Lin, K., Lin, C.: A study on reduced support vector machines. IEEE Transactions on Neural Networks **14**(6) (2003) 1449–1459

[13] Chan, P., Stolfo, S.: Toward parallel and distributed learning by meta-learning. AAAI Workshop in Knowledge Discovery in Databases (1993) 227–240

[14] Lazarevic, A., Obradovic, Z.: Boosting algorithms for parallel and distributed learning. Distributed and Parallel Databases **11**(2) (2002) 203–229

[15] Schapire, R.E.: The strength of weak learnability. Machine Learning **5** (1990) 197–227

[16] Chawla, N.V., Hall, L.O., Bowyer, K.W., Moore, T.E., Kegelmeyer, W.P.: Distributed pasting of small votes. Multiple Classifier Systems (2002) 52–61

[17] Siersdorfer, S., Sizov, S.: Automatic document organization in a P2P environment. European Conference on Information Retrieval (2006) 265–276

[18] Gorodetskiy, V., Karsaev, O., Samoilov, V., Serebryakov, S.: Agent-based service-oriented intelligent P2P networks for distributed classification. International Conference on Hybrid Information Technology (2006) 224–233

[19] Luo, P., Xiong, H., Lü, K., Shi, Z.: Distributed classification in peer-to-peer networks. Knowledge Discovery and Data Mining (2007) 968–976

[20] Asuncion, A., Newman, D.: UCI machine learning repository (2007)

[21] Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. IEEE **86**(11) (1998) 2278–2324

[22] Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001) Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.