Cascade RSVM in Peer-to-Peer Networks

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

Nanyang Technological University, Singapore

Abstract. The goal of distributed learning in P2P networks is to achieve results as close as possible to those from centralized approaches. Learning models of classification in a P2P network faces several challenges like scalability, peer dynamism, asynchronism and data privacy preservation. In this paper, we study the feasibility of building SVM classifiers in a P2P network. We show how cascading SVM can be mapped to a P2P network of data propagation. Our proposed P2P SVM provides a method for constructing classifiers in P2P networks with classification accuracy comparable to centralized classifiers and better than other distributed classifiers. The proposed algorithm also satisfies the characteristics of P2P computing and has an upper bound on the communication overhead. Extensive experimental results confirm the feasibility and attractiveness of this approach.

1 Introduction

Peer-to-peer (P2P) network is a large network of entities interconnected in a point-to-point manner. The Internet as a large point-to-point network of computers is a P2P network. P2P computing refers to computations performed in a P2P network of computers where there is no absolute centralized control. In recent years, data mining in P2P networks has attracted much attention as increasingly many applications have distributed data, from which useful knowledge may be mined. For instance, clustering and classification of peer data may reveal networks of cliques in social networks, and classification of network traffic could provide valuable information about network intrusions or usage behaviors.

The primary goal of learning in a P2P network is to achieve learning result that is as close as possible to that of a centralized approach. Learning models of classification (also clustering) is faced with several challenges [6]. In a P2P setting, learning algorithms need to take into account the scalability issue (Can the algorithm be computed when there are millions of peers?), peer dynamism (Can the algorithm deal with the availability and unavailability of data as peers connect and disconnect from the network?), asynchronism (Can the algorithm produce sufficiently accurate results without global synchronization?), and data privacy (Can the algorithm preserve the privacy of peer data when learning the global model?).

In this paper, we study the feasibility of learning in a P2P network in the context of learning classifiers. In particular, we are interested to know how Support

W. Daelemans et al. (Eds.): ECML PKDD 2008, Part I, LNAI 5211, pp. 55–70, 2008.

Vector Machines (SVM) perform in a P2P network, as SVM is a class of powerful classification and regression algorithms. Some of the weaknesses of SVM is in its memory and computational requirements, which increase with the size of training data set. A proven approach for alleviating such requirements, while not degrading classification performance, is the cascade SVM approach, where the data set is partitioned into smaller chunks and small-scale SVM learning is performed on these chunks. Support vectors obtained from these small-scale SVM models are combined with other chunks to derive better support vectors and improve the final set of support vectors.

We show how the cascading of SVM learning can be mapped to a P2P network of data propagation. As network communication is a performance issue, we show how the sending of support vectors from peer to peer can be improved using the Reduced Support Vector Machine (RSVM) [13] approach. Our proposed P2P SVM provides a method for learning classifiers in a P2P network that has classification accuracy comparable to a centralized classifier, yet satisfies the characteristics of P2P computing and has an upper bound on the communication overhead. We have implemented P2P SVM and experimental results confirm the feasibility and attractiveness of using this approach.

We focus mainly on the classification accuracy, scalability in terms of computation and total bandwidth usage, effects of data distribution and imbalanced class distribution. The other issues affecting classification in P2P networks such as peer dynamism, data privacy, security and different types of P2P networks will be studied in future works.

Our contributions in this paper are as follows: (1) We demonstrate the feasibility of cascade SVM in a P2P network, which to the best of our knowledge, is the first such attempt. (2) Our proposed P2P SVM has classification accuracy comparable to a centralized solution and better than other classification approaches in a P2P network. (3) In order to reduce data propagation cost in a P2P setting, we show how an upper bound can be derived to control the network communication overhead.

The organization of this paper is as follows: Section 2 describes related work. Section 3 introduces our proposed approach to perform SVM in a P2P network. Section 4 describes our experiments and the last section concludes the paper.

2 Background and Related Work

Many well-known classifiers such as decision tree, nearest neighbor classifier, artificial neural networks, Bayes classifier and support vector machine (SVM) work well with small datasets, but fail to maintain reasonable time and cost benefits on large datasets common to many domains. Hence, researchers have developed alternative methods such as selective sampling [2, 13, 14], parallelized and distributed learning [3, 9, 12, 15, 19, 23], in order to learn from such large datasets.

Breiman [2] introduced the pasting of Ivotes (or Rvotes) that trains an ensemble of classifiers, each built from a subset of data that has been selectively

sampled using out-of-bag estimation (or randomly sampled for Rvotes). Lee and Mangasarian [13] presented the Reduced Support Vector Machines (RSVM) approach that solves the SVM optimization problem using a randomly selected smaller portion of the whole dataset. Lin and Lin [14] studied several implementations of RSVM and showed that for problems with dense support vectors, RSVM significantly reduces the training time that is required for a small drop in accuracy compared with the SVM solution.

Parallelized and distributed algorithms represent another paradigm to solve the large dataset problem. These algorithms can be broadly classified as ensemble and cascade approaches. In general, these approaches split a large problem into smaller easily solvable sub-problems, and then combine their results. A useful side-effect of approaches under this paradigm is that they can also be used on *naturally distributed* data, saving the cost of moving data to a single location for training using a centralized solution.

Distributed ensemble approaches can be further divided as voting [5, 12] and meta-learning [3, 7, 17] approaches. Voting approaches build an ensemble of classifiers and then perform final classification based on the votes of all classifiers in the ensemble. For instance, Lazarevic and Obradovic [12] provide a distributed boosting framework that exchanges training statistics and performs weighted majority voting to obtain the final prediction. On the other hand, Chawla *et al.* [5] present a distributed version of Ivotes (DIvotes) and Rvotes (DRvotes) that works by first splitting the data, then performing Ivotes on each subset, and obtaining the final hypothesis by majority voting. The advantage that DIvotes has over distributed boosting is that no communication is required among the distributed parities during the training phase, thus, significantly reducing the communication overhead.

Meta-learning is in essence the learning of meta-attributes generated from various learners (classifiers). Chan and Stoflo [3] present an arbiter tree approach that builds various levels of classifiers and combines the results using arbitration rules. More recently, Pfahringer *et al.* [17] proposed to landmark various learning algorithms in order to characterize the classification problems and find the relationship between classifiers, whereas Džeroski and Ženko [7] presented their approach of using the model tree induction to learn the meta-level features generated.

Cascade learning was proposed mainly for the purpose of speeding up computation. Tveit and Engum [19] pioneered the work on cascading SVM by providing a heap-based tree topology framework for parallelizing the computation of Proximal SVM. Since then, a number of works have focused on cascade SVM [9, 15, 23]. Lu *et al.* [15] presented and compared various ways of cascading SVM. Zhang *et al.* [23] further improved cascade SVM by examining various ways of performing feedback to obtain a global optimal solution. Graf *et al.* [9] also provided a cascade SVM algorithm with feedback and formally proved the convergence of the algorithm.

2.1 Learning in P2P Networks

In recent years, there has been increasing interest in classification problems in P2P networks. A P2P network consists of a set of k peers $P = \{p_1, \ldots, p_k\}$, where all peers function equally as both servers and clients. However, a P2P environment possesses unique characteristics that introduce challenges for the classification task. These characteristics include scalability, peer dynamism, data dynamism, asynchronism and privacy and security [6]. P2P networks can be considered as a massively distributed environment as the number of peers, k, in the network usually exceeds hundreds or thousands. In addition, these peers may leave and join the network anytime, and the data they possess may change frequently. Due to the size of the network, it is not feasible to perform synchronization considering the network latency and bandwidth. If data exchange among peers is involved, privacy and security may also pose concerns.

Based on data propagation, existing P2P classification approaches can be categorized as 1) model propagation [18] and 2) test data propagation [8, 16] approaches. Model propagation approaches build local classifiers on each peer and then propagate the model to other peers. The peers can then use the collected models for performing classification. In the latter approach, a peer only propagates test instances to other peers, which in turn classify these instances and return results to the requesting peer. The model propagation approaches generally incur more communication cost during the model construction phase, which exacerbates when the classification model frequently changes. However, under this approach, classification of test instances is faster and peers have more freedom on how the models can be used (e.g., perform meta-learning using the models).

Siersdorfer and Sizov [18] have proposed a framework for classifying web documents in a P2P environment. The algorithm trains a local classifier and propagates it to other peers. Each peer then uses the received models to construct a meta model for performing classification. Although the paper states that the propagated model should be a compressed representation of the local data set, it neither provides details on how this may be achieved, nor on how the models may evolve with the addition of new data. Furthermore, the tuning of the global model to improve accuracy requires synchronization among peers, which increases communication cost.

On the contrary, test instance propagation approaches are not affected by frequent changes of models and does not incur communication cost during construction of models. However, classification tasks are slower since requests have to be made to the P2P network, and if these tasks are frequent, the communication cost can be comparable to that of the model propagation.

Gorodetskiy *et al.* prototyped an agent-based, service-oriented P2P distributed classification approach [8]. However, the focus of the paper is not on the classification task, but to provide a proof-of-concept implementation and to explore the issues that may exist in the agent-based, service-oriented P2P network.

More recently, Luo *et al.* [16] proposed a P2P classification approach by pasting of small votes. In this approach, each peer pastes small bites to build local

classifiers until the error between subsequent models falls below a certain threshold. The final classification is then performed by sending classification requests to all peers based on an optimal communication protocol.

3 Approach

In this section, we present our proposed approach, illustrating the design process and finally providing a complexity analysis. Our approach based on the cascade SVM paradigm, is specifically designed for the P2P network, addressing the additional constraints not found in the general distributed and parallel computing environment. The three basic processes in cascade SVM are: 1)build an SVM for each of the local data, then iteratively 2) propagate and 3) merge the models to create an improved SVM until all subsets have been combined. Let us now examine cascade SVM and our proposed approach in detail.

3.1 Cascade SVM

In cascade SVM, the algorithm starts by building SVM using local data. The purpose of using SVM (as well as merging) is to filter out as many non support vectors as early as possible, to reduce the time and space complexity required to efficiently build the global solution. However, using standard SVM may generate quite a high number of support vectors. Since our approach requires propagation of models in the P2P network, these large number of support vectors result in a high communication cost. Hence, algorithms based on standard SVM are usually not viable. Therefore, our criteria for building local classifiers changes from being able to effectively filter out redundant data, to being able to extract a very small set of representative data.

3.2 P2P Cascade RSVM

Based on the above considerations, we employ an approximate SVM solution - RSVM, which reduces the number of support vectors, for the task. The disadvantage of using RSVM is that the resulting cascade SVM cannot produce a global optimal solution. By global optimal solution, we refer to the solution produced by SVM and cascade SVM with feedback/synchronization, which however, is infeasible to achieve, since the convergence to the global optimal solution requires synchronization among all peers (for the validation process). As the number of support vectors in a SVM has extensive influence on the memory and training time, being able to reduce the number of support vectors greatly improves the training speed and lowers the memory requirements. However, since the SVM decision hyperplane is constructed from these support vectors, reducing the number of support vectors may also reduce the classification accuracy. Despite this, it has been found that RSVM can use a very small subset to represent the whole data, with only a slight drop in classification accuracy compared to traditional SVM [16]. Hence, usage of RSVM does not cause any serious drawback.

Since peer data constantly change in a P2P network, a set of new training data is treated as a new peer's dataset, and goes through the same processes as the existing local data. This addresses the data dynamism issue, allowing incremental learning. Although, our approach allows incremental learning, decremental learning or removal of data is not addressed, as this concerns the issue of concept drift and is not within the scope of this paper.

After the model is generated, it is propagated to other peers. Despite the main disadvantage of high communication cost (effect reduced as stated above), model propagation provides a way to counter the peer dynamism constraint. With model propagation, even when peers go offline, their models still exist on other peers on the P2P network (provided they have successfully propagated to other peers before they went offline). This allows sharing of models between peers which were not present on the P2P network at the same time, which is an important factor for maintaining high classification accuracy within the P2P network. In addition, our approach ensures that models are only collected/merged once to prevent duplication. Besides these, model propagation guarantees achieving a local optimal solution with cascade SVM, since it becomes possible to validate using the peers' models, and the high duplication rate of models allows higher throughput for the transfer of models.

Similar to the automatic document organization approach, model propagation in our approach can be implemented separately from the building of the classifier. This allows our approach to be deployed in any type of P2P network increasing its flexibility. By viewing the models as files in a P2P network, we can map the problem of model propagation in P2P network to the file propagation problem in P2P network, which has been extensively studied. For our approach, we utilize the UPTReC [21] algorithm, because it provides a probabilistic guarantee in file consistency which helps to ensure that models can be properly propagated within the P2P network. Experiments [21] show that UPTReC can reduce up to 70% overhead messages compared with other existing techniques.

Models are collected as peers propagate them in the P2P network. In contrast to the cascade SVM, since we do not have control over how, when and how many of the peers' models will be collected, we perform the merging process as follows. All models collected within t duration are merged together in a single process and then merged with the peer's local optimal SVM. In the two extreme cases, given $t = 0$, this simply implies that each time a peer's model is collected, it is merged immediately with the last cascaded model, and given $t =$ the time required to collect models of all uncollected peers in the P2P network, all newly collected models are merged in a single process with the previously cascaded RSVM. For example, consider that peer j receives three new models from other peers before time t after startup, and two other new models between time t and $2t$. Therefore, at time t from startup, peer j will merge the three newly received models with the latest local model, and at time $2t$ from startup, it will merge the two newly received models with the latest cascaded model. This process is illustrated in Figure 1, and the training phase of the proposed approach is given in Algorithm 1.

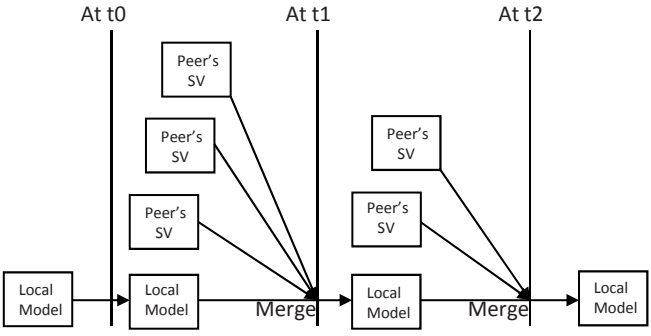


Fig. 1. Illustration of merging support vectors

To summarize, the main differences between existing cascading approaches and our proposed P2P Cascade RSVM lies in the use of RSVM, and in the ad-hoc merging of the collected models due to the high dynamism of the P2P networks. This greatly reduces the communication overhead for distributing data after distributed and parallelized construction of local models. These extensions of cascade SVM make it feasible to learn from the P2P environments and even achieve results comparable to centralized solution, while reducing computation and communication costs.

Algorithm 1. P2P Cascade RSVM algorithm for peer p_i

input: the percentage p of support vectors to use,
the duration t to wait before merging,
local training data D_i

$SSV_i = \{\}$
$PSV_i = \{\}$
training data $T = \emptyset$
Train local classifier model M_i using RSVM on D_i
Propagate the support vectors SV_i of M_i to other peers
while *true* **do**
  while *waiting time* $< t$ **do**
    foreach SV_j *of peer* p_j *received* **do**
      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 Model Propagation Cost

In our approach, since the number of support vectors directly determines the size of the model to be propagated, the communication cost can also be greatly reduced. Furthermore, by specifying the size of the support vectors, either absolutely or as a percentage of the training data, we can give an upper bound on the communication cost of the construction of the cascade SVM as follows. Let N be the total number of peers in the P2P network, l be the total size (in terms of number of vectors) of the problem and $s, s < 1$ be the percentage of the problem to be used as support vectors. Then the upper bound of the total communication cost, c, required for all peers to obtain the global model is

$$c = N \cdot l \cdot s \tag{1}$$

for a two-class problem. For a multi class problem, where the number of classes is nc, and using the one-against-one strategy for SVM classification, the cost is as follows:

$$c = N \cdot l \cdot s \cdot (nc \cdot (nc - 1)/2) \tag{2}$$

3.4 Computation Cost

Considering the following SVM optimization problem [20]:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2} w^T w + C(\sum_{i=1}^{l} \xi_i^2) \\ \text{subject to} \quad & y_i(w^T w \phi(x_i) + b) \geq 1 - \xi_i \end{aligned} \tag{3}$$

given that x_i is a feature vector and y_i is the corresponding label of a training set, where $x_i \in R^n$ and $y_i \in \{1, -1\}$. As $\phi(x)$ maps x into a higher dimensional space, we can simply solve its dual, which is a quadratic programming problem:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \alpha^T (Q + \frac{I}{2C}) \alpha - e^T \alpha \\ \text{subject to} \quad & y^T \alpha = 0, \\ & 0 \leq \alpha_i, i = 1, \ldots, l \end{aligned} \tag{4}$$

where the number of variables equals l, e is the unity vector, Q is an l by l positive semi-definite matrix, $Q_{ij} \equiv y_i y_j K(x_i, x_j)$ and $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$ is the kernel function. Computing the kernel function $K(x_i, x_j)$ for every training instance costs $O(l^2)$ and solving (4) costs $O(l^3)$.

However, for RSVM based on Least-Square SVM, we are only required to solve

$$\min_{\tilde{\alpha}} f(\tilde{\alpha}) = \frac{1}{2C} \tilde{\alpha}^T \tilde{\alpha} + \tilde{\alpha}^T (\tilde{Q}^T \tilde{Q}) \tilde{\alpha} - 2 e^T \tilde{Q} \tilde{\alpha} + e^T e \tag{5}$$

where f can minimized by finding the solution of $\frac{\partial f}{\partial \tilde{\alpha}_i} = 0, i = 1, \ldots, m$:

$$\frac{1}{C}\tilde{\alpha} + 2\tilde{Q}^T\tilde{Q}\tilde{\alpha} - 2\tilde{Q}Te = 0, \tag{6}$$

$$(\tilde{Q}^T\tilde{Q} + \frac{I}{2C})\tilde{\alpha} = \tilde{Q}^T e \tag{7}$$

a positive definite linear system of size m, where m is the size of the subset R used in RSVM, $\tilde{\alpha}$ are the coefficients of the separating hyperplane and $\tilde{Q} = [Q_{:,R} \quad y]$. Hence, the total time complexity for RSVM is $O(lm^2)$. For the complete formulation, refer to [14].

To analyze the time complexity of our approach, we have to examine the process of building the local model and merging of the collected models. Given a P2P network with N peers, and total training data of size l, let the size of local data for peer i be l_i, and the percentage of local data to be used for RSVM be $s, s < 1$. Then, the size of the subproblem to solve in RSVM is $m_i = l_i s$, and the time complexity for building a local model for peer i with RSVM is $O(l_i m_i^2)$. Since the size of the subproblem optimized by RSVM is already very small, and we have no prior knowledge of the amount of reduction that can be achieved by the optimization process, we assume that the size of the support vectors for the resulting models is the same as the size of the subproblem. Hence, after a peer constructs the local model (of size at most m_i), it propagates the model to other peers. The size of the support vectors collected from all peers is $m = \sum_1^N m_i$. If traditional SVM is used, the complexity of merging is $O(m^3)$. However, with other more efficient techniques such as SMO, cost of merging can be reduced, but in this case, we use the complexity of SVM to provide the upper bound. All in all, with $l_i m_i^2 << m^3$, the complexity of our proposed approach is $O(m^3)$.

Using centralized SVM and RSVM as comparison, we present a summary of the computation and communication costs of the various SVM based approaches in Table 1. It can be seen from Table 1 that our proposed approach has the least cost with respect to the centralized approaches.

Table 1. Summary of the training costs

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

4 Experiments and Result Analysis

Here, we present the experimental results on some large sized problems to simulate the problem size that may exist in a real P2P environment. First we describe the experimental setup. Then we compare the classification accuracy of centralized and existing P2P classification approaches, followed by a demonstration of

the effect of scalability, peers' data distribution and data class distribution on the various algorithms. Finally we illustrate the effect of the number of support vectors on the classification accuracy of our approach.

4.1 Experimental Setup

We used the covertype dataset and the waveform data generator available from the UCI repository [1]. For waveform, we generated 100,000 instances with 21 attributes. The covertype dataset was used further to generate a binary covertype dataset with class two versus all other classes. Summary of the datasets used is presented in Table 2. All attributes of the datasets were scaled to between -1 and 1.

Table 2. Summary of the datasets used in experiments

	Instances	Attributes	Classes
Binary Covertype	581,012	54	2
Covertype	581,012	54	7
Waveform	100,000	21	3

The experiments were conducted on a cluster of 16 machines, each with two Intel Dual Core Xeon 3.0GHz processors, 4 GB of Ram and connected by a gigabit ethernet.

The J48 algorithm (variant of the C4.5), from Weka [22] was used for the centralized classification and as the base classifier for the algorithm from [16]. In addition, we implemented the algorithm from [16], which we refer to as P2P Ivotes, in Java. We used the C-SVC algorithm from LIBSVM [4], in C++ as the centralized SVM solution, and used RSVM based on Least Square SVM algorithm from [14] in our approach which was implemented in C++.

In all P2P experiments, unless otherwise stated, we used 500 peers, and divided the data equally among them. We did not experiment with more peers since this would result in unrealistically small sizes for local peer data, which would adversely affect performance of the P2P approaches. For P2P Ivotes, bite size of 800 and λ of 0.02 were used. For SVM, SVM Ensemble and P2P Cascade RSVM, we used the RBF kernel, and for each dataset, the γ and C values were chosen using the model selection tool provided with LIBSVM on a 1 percent stratified sampled data of the whole dataset. For all datasets, we used 1 percent of the data as support vectors for our P2P cascade RSVM.

4.2 Classification Accuracy

In this experiment, we conducted a 10-fold cross validation using centralized RSVM, centralized J48, plurality voting on ensemble of J48, plurality voting on ensemble of SVM, P2P Ivotes and P2P Cascade RSVM on the binary covertype, covertype and waveform dataset. In order to train all peers on the same amount

Table 3. Tenfold cross-validation results

Dataset	Accuracy (%)					
	RSVM	J48	J48 Ensemble	SVM Ensemble	P2P Ivotes	P2P Cascade RSVM
Binary Covertype	71.97	**73.3**	58.26	54.16	58.72	72.93
Covertype	**68.16**	66.59	54.47	44.65	54.14	67.77
Waveform	99.8	**99.86**	99.62	99.79	99.78	99.61

Table 4. Average training time

Dataset	Time (secs)					
	RSVM	J48	J48 Ensemble	SVM Ensemble	P2P Ivotes	P2P Cascade RSVM
Binary Covertype	111.2	2357.9	159.04	48	326.56	**11.9**
Covertype	751.8	2501.64	238.12	**53.5**	378.38	126.4
Waveform	32	13.8	12	6.2	12	**0.4**

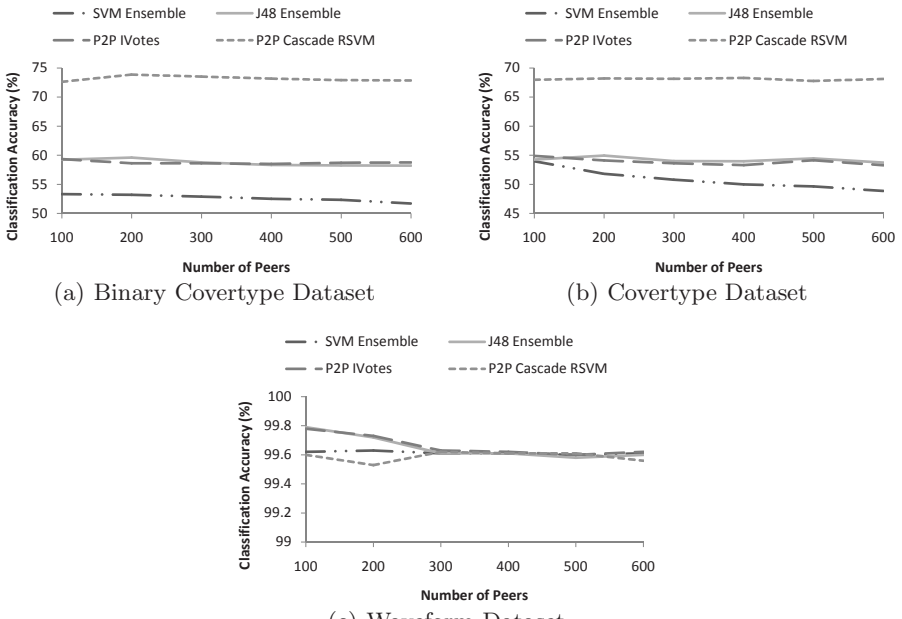
of data, we used 500 peers for binary covertype and covertype datasets and 100 peers for the waveform dataset. The classification accuracy and average training time taken are shown in Tables 3 and 4 respectively.

As shown in Table 3, our approach has accuracy comparable to the centralized solution on all datasets. Compared with other existing approaches, our approach exhibits similar accuracy on the waveform dataset, but has far better accuracy on the binary covertype and covertype datasets. In addition, our approach has the least training time for binary covertype and waveform dataset and second least for covertype dataset, which is probably due to the higher number of classes in the latter. We note that the P2P Ivotes results obtained by our experiments are dissimilar to those reported in [16], perhaps due to different methods of assigning peers' local training sets.

4.3 Scalability

To determine the scalability of the various P2P classification approaches, we varied the number of peers from 100 to 600 based on a 10-fold cross validation. For all approaches, the training data is divided equally among all peers with random class distribution.

As can be seen in Figure 2, our approach achieves significantly (based on student's t-test with p-value of 0.05) higher accuracy on the binary covertype and covertype data while producing similar accuracy on the waveform dataset. We observe in Figure 2(a) and 2(b), that the two covertype datasets show similar results, which is not surprising. For the Waveform dataset, none of the approaches seem to be affected by the number of peers that exist in the network. However, for both covertype datasets, all approaches except ours lose some accuracy when the number of peers increases. It is also noted that for all the datasets, the results of the J48 ensemble and the P2P Ivotes showed similar trends.



(a) Binary Covertype Dataset

(b) Covertype Dataset

(c) Waveform Dataset

Fig. 2. Effect of P2P network size on accuracy

We observed that some of the random data assignment resulted in a few peers not obtaining data from certain classes, which might explain the poor performance of the ensemble methods. This hypothesis will be verified in future work with further experiments.

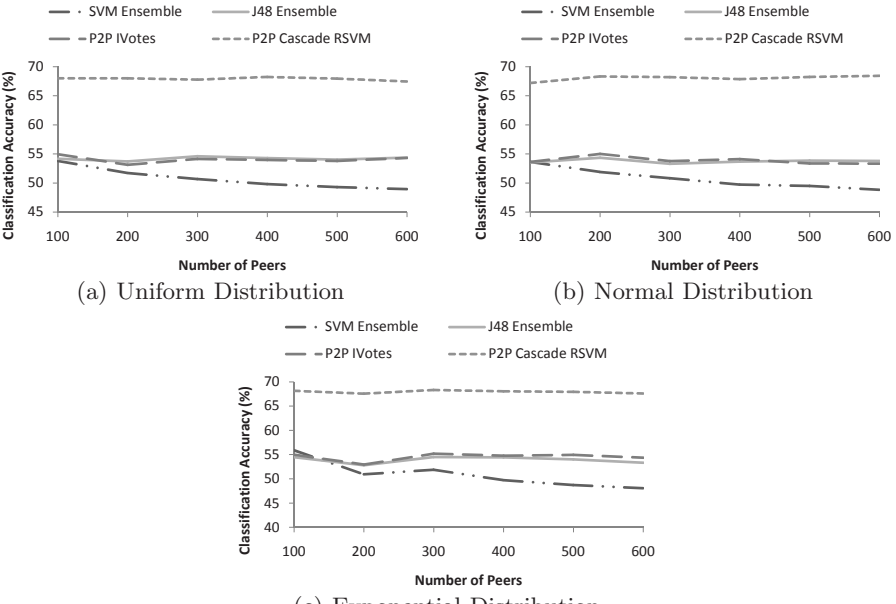
4.4 Peers' Data Distribution

Here we illustrate the effect of distribution of peers' data on the classification accuracy. From 100 to 600 peers, we randomly assign a subset of the data to each peer, where the size of the subset is based on exponential, uniform and normal distributions and test the accuracy using 10-fold cross validation. We have used the covertype dataset in this experiment.

As observed in Figure 3, the results for the different distributions do not seem to be very much different. Including the results from Figure 2(b), which is based on equal distribution, we conducted a student's t-test and found that there is actually no significant difference for each algorithm between the results of the different distributions. However, it would be interesting to see how peers dynamism can actually affect accuracy based on the different data size distribution.

4.5 Effect of Imbalanced Class Distribution

To see if the P2P classification approaches can deal with peers having data with imbalanced class distribution (natural class distribution of the whole dataset



(a) Uniform Distribution (b) Normal Distribution

(c) Exponential Distribution

Fig. 3. Effect of peers' data distribution on accuracy (covertype dataset)

remains unchanged), we purposely vary the class distribution of the data subset assigned to each peer. Using the binary covertype data, we modify the class distribution such that the class distribution has d percentage of skew compared to the natural class distribution. For example, if the natural class distribution is 60/40, a skew of $d = 10\%$ generates a modified class distribution of 65/35 for half of the peers and 55/45 for the other half of the peers. Although we modified the class distribution of the local training data, we still ensured that every peer received the same amount of data.

The results in Figure 4 show that our approach achieves better accuracy in the presence of imbalanced class distribution. Performing a student's t-test shows that the difference in accuracy between the other existing P2P approaches and our approach is significant with p-value of 0.05. Note that with the increase in percentage of skewness, the accuracy of the J48 ensemble and P2P Ivotes gradually decreases. However, the accuracies of our approach and SVM ensemble are not affected. Our approach is unaffected by the class imbalance perhaps due to the merging of support vectors that may have a rebalancing effect on the class distribution.

4.6 Size of Support Vectors

By restricting the number of support vectors used to build the SVM, we can limit the communication, computation and memory cost, albeit possibly at the

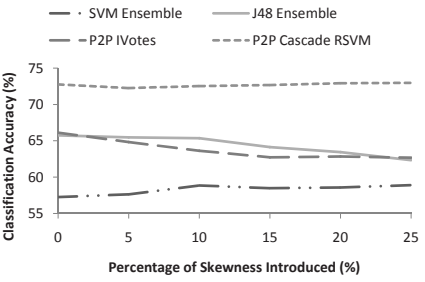


Fig. 4. Effect of imbalance class distribution on accuracy (binary covertype dataset)

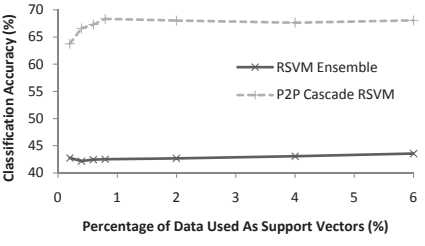


Fig. 5. Effect of support vector size constraint on accuracy (covertype dataset)

expense of classification accuracy. Here, we demonstrate the effect of the number of support vectors used on the accuracy. This experiment was conducted on the covertype data using an ensemble of RSVM and P2P cascade RSVM with 500 peers, by varying the percentage of support vectors used (subproblem size) from 0.2 to 6 percent.

From Figure 5, we note that when the percentage of support vectors used is too small (i.e. less than 1 percent), the classification accuracy is not stable and of unacceptable level. However, when the percentage of support vectors increases to above 1 percent, the increase in accuracy starts to plateau. Another point to note is that time and memory complexity of SVM is quadratic with respect to the number of support vectors. Therefore a low percentage of support vectors would be preferred but care must be taken to ensure that there are enough support vectors to represent the peers' local training data (which is dependent on the size of local training dataset).

5 Conclusion

In this paper, we study the problem of learning models of classification in a P2P network. We have proposed a combination of the cascade SVM and Reduced Support Vector approaches to learn classifiers in a P2P setting. Experimental results show that our proposed approach can learn classifiers with accuracies close to those of centralized approaches. Moreover, our approach also outperforms other

distributed models of classifier learning. The proposed approach scales with the size of the network, and accuracy is not affected by the number of peers. Also, we provide an upper bound on the massive communication overhead in P2P classification using the Reduced Support Vector approach to cap the number of support vectors computed. Overall, experimental results confirm the feasibility and attractiveness of using our approach. As part of future work, we will be exploring in detail, the effects of peer dynamism, cliques, and data privacy on the problem of learning in P2P networks. In addition, we will investigate unified kernel machines [11] and distributed active learning [10] techniques for enhancing classification performance.

References

[1] Asuncion, A., Newman, D.: UCI machine learning repository (2007)
[2] Breiman, L.: Pasting small votes for classification in large databases and on-line. Machine Learning 36(1-2), 85–103 (1999)
[3] Chan, P., Stolfo, S.: Toward parallel and distributed learning by meta-learning. In: AAAI Workshop in Knowledge Discovery in Databases, pp. 227–240 (1993)
[4] Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm
[5] Chawla, N.V., Hall, L.O., Bowyer, K.W., Moore, T.E., Kegelmeyer, W.P.: Distributed pasting of small votes. In: Multiple Classifier Systems, pp. 52–61 (2002)
[6] 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), 18–26 (2006)
[7] Džeroski, S., Ženko, B.: Is combining classifiers with stacking better than selecting the best one? Machine Learning 54(3), 255–273 (2004)
[8] Gorodetskiy, V., Karsaev, O., Samoilov, V., Serebryakov, S.: Agent-based service-oriented intelligent P2P networks for distributed classification. In: International Conference on Hybrid Information Technology, pp. 224–233 (2006)
[9] Graf, H.P., Cosatto, E., Bottou, L., Dourdanovic, I., Vapnik, V.: Parallel support vector machines: The cascade SVM. In: NIPS (2004)
[10] Hoi, S.C.H., Jin, R., Zhu, J., Lyu, M.R.: Batch mode active learning and its application to medical image classification. In: ICML, pp. 417–424 (2006)
[11] Hoi, S.C.H., Lyu, M.R., Chang, E.Y.: Learning the unified kernel machines for classification. In: SIGKDD, pp. 187–196 (2006)
[12] Lazarevic, A., Obradovic, Z.: Boosting algorithms for parallel and distributed learning. Distributed and Parallel Databases 11(2), 203–229 (2002)
[13] Lee, Y., Mangasarian, O.: RSVM: Reduced support vector machines. In: SIAM International Conference on Data Mining, pp. 00–07 (2001)
[14] Lin, K., Lin, C.: A study on reduced support vector machines. IEEE Transactions on Neural Networks 14(6), 1449–1459 (2003)
[15] Lu, B., Wang, K., Wen, Y.: Comparison of parallel and cascade methods for training support vector machines on large-scale problems. In: International Conference on Machine Learning and Cybernetics, pp. 3056–3061 (2004)
[16] Luo, P., Xiong, H., Lü, K., Shi, Z.: Distributed classification in peer-to-peer networks. In: SIGKDD, pp. 968–976 (2007)
[17] Pfahringer, B., Bensusan, H., Giraud-Carrier, C.G.: Meta-learning by landmarking various learning algorithms. In: ICML, pp. 743–750 (2000)

[18] Siersdorfer, S., Sizov, S.: Automatic document organization in a P2P environment. In: European Conference on IR Research, pp. 265–276 (2006)
[19] Tveit, A., Engum, H.: Parallelization of the incremental proximal support vector machine classifier using a heap-based tree topology. Technical report, IDI, NTNU, Trondheim, Norway (2003)
[20] Vapnik, V.N.: The nature of statistical learning theory. Springer, New York (1995)
[21] Wang, Z., Das, S.K., Kumar, M., Shen, H.: An efficient update propagation algorithm for P2P systems. Computer Communications 30(5), 1106–1115 (2007)
[22] Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
[23] Zhang, J., Li, Z., Yang, J.: A parallel SVM training algorithm on large-scale classification problems. In: International Conference on Machine Learning and Cybernetics, pp. 1637–1641 (2005)