Satrap: Data and Network Heterogeneity Aware P2P Data-Mining[*]

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

Nanyang Technological University, Singapore

Abstract. Distributed classification aims to build an accurate classifier by learning from distributed data while reducing computation and communication cost. A P2P network where numerous users come together to share resources like data content, bandwidth, storage space and CPU resources is an excellent platform for distributed classification. However, two important aspects of the learning environment have often been overlooked by other works, viz., 1) location of the peers which results in variable communication cost and 2) heterogeneity of the peers' data which can help reduce redundant communication. In this paper, we examine the properties of network and data heterogeneity and propose a simple yet efficient P2P classification approach that minimizes expensive inter-region communication while achieving good generalization performance. Experimental results demonstrate the feasibility and effectiveness of the proposed solution.

keywords: Distributed classification, P2P network, cascade SVM.

1 Introduction

P2P networks contain large amounts of data naturally distributed among arbitrarily connected peers. In order to build an accurate global model, peers collaboratively learn [1,2,3,4] by sharing their local data or models with each other. Though recent efforts aim to reduce this communication cost compromise, none of them take into account heterogeneity in either the network or the data.

In order to build a global model representative of the entire data in the P2P network, only dissimilar data (from different data subspaces) need to be shared. While sharing similar data (from the same data subspace) adds no value to the global model, it only adds to the communication cost which can be prohibitive if the data were from geographically distant peers.

In this paper, we address the problem of learning in a P2P network where data are naturally distributed among the massive number of peers in the network. In addition, the location of these peers span across a large geographical area where *distant peers incur higher communication cost* when they try to communicate. Moreover, there is a possibility that the *data of different peers overlap in the*

[*] This work is partly supported by HP Labs Innovation Research Program 2008 grant.

M.J. Zaki et al. (Eds.): PAKDD 2010, Part II, LNAI 6119, pp. 63–70, 2010.
© Springer-Verlag Berlin Heidelberg 2010

problem space. An approach that simply exchanges data of all peers will incur a high communication cost in order to achieve high accuracy. On the other hand, an approach that does not exchange data will achieve low prediction accuracy in order to save communication costs. Hence, the objective would be to achieve the best global accuracy-to-communication cost ratio.

In this paper, we describe a data and network heterogeneity aware adaptive mechanism for peer-to-peer data-mining and study the relationship between the training problem space and classification accuracy. Our proposed approach, Satrap,

- achieves the best accuracy-to-communication cost ratio given that data exchange is performed to improve global accuracy.
- allows users to control the trade-off between accuracy and communication cost with the user-specified parameters.
- is insensitive to the degree of overlapping data among peers.
- minimizes communication cost, as the overlapping data among different regions increase.
- is simple, thus making it practical for easy implementation and deployment.

2 Background and Related Work

A P2P network consists of a large number of interconnected heterogeneous peers, where each peer holds a set of training data instances. The purpose of classification in P2P networks is to *effectively* learn a classification model from the training data of all peers, in order to accurately predict the class label of unlabeled data instances.

Existing P2P classification approaches typically either perform local [5] or distributed [1,2,4] learning. Local learning performs training within each peer without incurring any communication between peers during the training phase. Luo *et al.* [5] proposed building local classifiers using Ivotes [6] and performed prediction using a communication-optimal distributed voting protocol. Unlike training, the prediction process requires the propagation of unseen data to most, if not all peers. This incurs a huge communication cost if predictions are frequent. On the contrary, instead of propagating test instances, the approach proposed by Siersdorfer and Sizov [4] propagates the linear SVM models built from local data to neighboring peers. Predictions are performed only on the collected models, which incur no communication cost.

Distributed learning approaches not only build models from the local training data, but also collaboratively learn from other peers. As a trade-off to the communication cost incurred during training, the cost of prediction can be significantly reduced. In a recent work, Bhaduri *et al.* [2] proposed an efficient approach to construct a decision tree in the P2P network. Over time, the induced decisions of all peers converge, and as the approach is based on distributed majority voting protocol, it incurs a lower communication cost compared to broadcast based approaches.

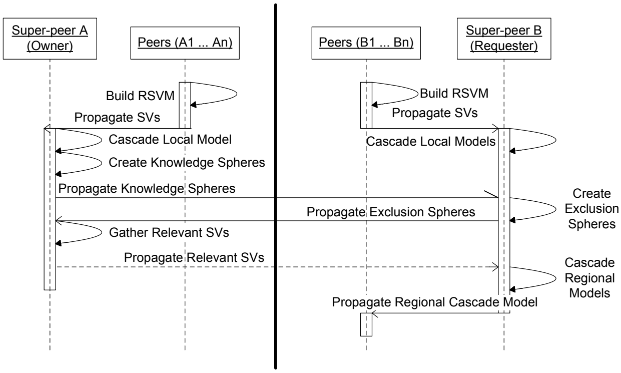


Fig. 1. Sequence diagram of Satrap (among two clusters of peers)

To reduce communication cost and improve classification accuracy, Ang *et al.* [1] proposed to cascade the local RSVM models of all peers (AllCascade). RSVM was chosen as it significantly reduces the size of the local model. However, AllCascade requires massive propagation of the local models and the cascading computation is repeated in all peers, wasting resources due to duplications.

3 Approach

Figure 1 depicts the process for constructing a global classification model in Satrap between two clusters of peers (i.e., communications are performed in a pairwise manner between different regions). Rather than flooding the entire network with models (as in AllCascade), here each peer builds an RSVM on its local data, and propagates it only within its own geographic region. This is feasible as intra-region communication is inexpensive.

Then one distinguished peer is elected from each region as the super-peer, which combines (and compresses) the models received into a regional model, and transfers them to other regions through their respective super-peers. These super-peers serve as a single point of communication between regions[1], thus reducing expensive inter-regional communication. However, note that the use of super-peers doesn't lead to a single point of failure, since if one fails, another peer from the same region can be dynamically assigned with location aware P2P overlay networks [7]. The super-peer may also delegate actual communication tasks to other peers for better load-balancing.

Here, we have another innovation to further reduce this cost. Instead of receiving all models from other regions, each regional super-peer requests for certain models only. This is accomplished as follows. Every super-peer clusters its data, and sends its cluster information (called Knowledge Spheres, c.f. Section 3.1) to other super-peers. With this knowledge, each super-peer determines the overlap in underlying data space (called Exclusion Spheres, c.f. Section 3.2) between itself

[1] Hence the name Satrap - title for the governor of regional provinces in ancient Persia.

and others, and requests only for models from non-overlapping spaces from an owner super-peer. Upon receiving a request, the owner super-peer gathers support vectors (c.f. Section 3.3) from its model that are relevant to the requester's Exclusion Spheres, and transfers them.

Finally, each super-peer combines all the models received (as before) and then propagates the resultant model to all the peers within its region (c.f. Section 3.4), again with low intra-region cost.

Though this process requires communication of the compact data space representation between regional super-peers, it significantly reduces the propagation of models. In this paper, we omit detailed discussion on failure tolerance and load distribution, and limit our scope to only the data-mining related issues.

3.1 Knowledge Sphere Creation

Unlike test instance propagation where information cannot be compressed or filtered, model propagation in general, allows some form of compression or filtering while enabling representative global models to be constructed.

Since the models of super-peers from different geographical regions may be built on similar data (or data from the same data space), while creating a global model, it is unnecessary for a super-peer to receive *all* information from others. As we do not know a priori what data are overlapping between them, we need a representation of every super-peer's underlying data in the problem space. For this purpose, we propose the use of high dimensional sphere, created from clustering of the data.

After a super-peer cascades the models from its regional peers, we separate the support vectors (SVs) into their separate classes and cluster them. The separation allows more compact clusters to be generated, as SVs from different classes may lie in slightly different input space. The knowledge of these clusters, called the Knowledge Spheres, comprising the centroid (mean of all SVs), radius (maximum distance of any SV in cluster to the centroid), and their density (number of SVs within the cluster definition) is then propagated to all other super-peers.

The reason for using clustering is that it creates groups of neighboring data points which reside close to each other in the problem space, as represented by the high dimensional spheres. Although spheres may not represent the data as well as some other high dimensional shapes such as convex hulls or polygons, they are computationally cheapest to generate and have the best compression ratio (single centroid and radius). We have used agglomerative hierarchical clustering based on single linkage for this task, because it preserves the neighborhood information of the clusters. Another desirable property of this approach is that it produces deterministic results. We also use Euclidean distance as the distance measure for clustering, as it is shown to preserve the neighborhood property between input and feature space [8].

The clusters generated can affect the detection of (non) duplicated data, however we don't know a priori how many clusters would result in the most accurate detection of duplicates. Hence, instead of specifying the number of clusters, peers choose the desired cluster-to-SV ratio R, depending on how many support vectors they have. Note that as the number of clusters reduces, the input space covered by at least one cluster also increases in order to cover the points of the removed clusters. The increase in space covered also includes empty spaces. As the neighborhood area of the input space is correlated to the feature space [8], the feature space covered by the cluster also increases. If we were to filter from such a larger neighborhood (either input or feature space), more points potentially closer to the decision boundary would be filtered, leading to a possibly larger error. It is obvious that as heterogeneity of the regional data increases, the number of clusters required for a compact representation of the data also increases. Moreover, an increase in number of clusters always maintains or improves the cluster compactness (i.e., reduces the intra-cluster distance) but at the cost of addition communication overheard.

3.2 Exclusion Sphere Creation

When a super-peer (say, $r_{requester}$) receives another super-peer's (say, r_{owner}'s) knowledge spheres, it checks if it has as much knowledge about the data space as r_{owner}. It then informs r_{owner} of the knowledge it lacks, so that corresponding knowledge may be transferred. If the number of $r_{requester}$'s SVs falling within the space of an r_{owner} sphere is less than the density of the sphere (times a threshold T), $r_{requester}$ creates a exclusion sphere from those points. The information of the exclusion sphere (centroid, radius, density) along with the corresponding sphere that it overlapped with, is then sent to r_{owner} as part of the data request.

Note that this process is order-dependent. Once $r_{requester}$ has requested information from r_{owner} on a certain data space, it will not request information from another super-peer on an overlapping space, unless of course the latter has significantly larger density. We do not address the order dependency of overlapping checks due to several reasons. Firstly, in order to check the order, a super-peer has to wait for several super-peers to send their knowledge spheres, which is impractical in a dynamic P2P network. Secondly, order dependency only affects performance if there is a quality difference in the data of the different regions, but currently there is no way to verify this difference in quality (unless data points are sent for checking, which is what we want to avoid). Without additional knowledge on the data or communication cost, it would be infeasible to optimize the ordering.

3.3 Gather Relevant SVs

When r_{owner} receives the request, it chooses all SVs that are within the overlapping spheres but outside the exclusion spheres for transfer. It also chooses SVs that lie within the exclusion spheres with a probability of 1 - (number of SVs in exclusion sphere for $r_{requester}$ / number of SVs in exclusion sphere for r_{owner}). We use probabilistic sampling so that SVs within the exclusion sphere are chosen only when the confidence (number of SVs, evidence) of the $r_{requester}$ in the exclusion data space is lower than that of r_{owner}. All the chosen data points are then consolidated and sent to $r_{requester}$. This process marks the end of the cross region data probing and exchange. At this stage, $r_{requester}$ has received models from the entire network if it has requested from all other super-peers. Since the gathering of data is based on the clusters created from the local region cascaded model, it is not order-dependent.

3.4 Global Model Construction and Prediction

Once $r_{requester}$ receives the SVs from r_{owner}, they are merged with the SVs of the local regional cascaded model and the new global cascaded model is built. The new global model can be propagated down-stream to other local regional peers with cheap intra-regional communication. Since every peer now has the global model, all predictions can be made locally without incurring any extra communication cost. In addition, there is no need to wait for predictions from other peers which also saves time. With feedback proposed in [9], the incremental building of the global model at the super-peer is order invariant on the arrival of the exchanged models.

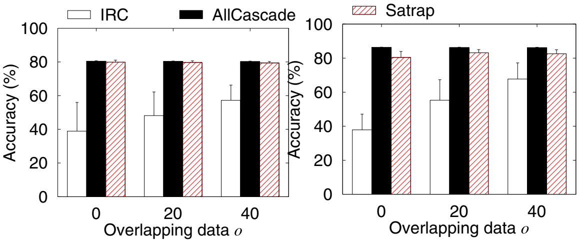
4 Experimental Results

Here, we demonstrate how Satrap exploits data heterogeneity to reduce communication overheads in presence of network heterogeneity, and achieves a good balance between accuracy and communication cost.

We used the multi-class Covertype (581,012 instances, 54 features, 7 classes and 500 peers) and multi-class Waveform (200,000 instances, 21 features, 3 classes and 100 peers) datasets [10]. The datasets were split into ten clusters, each assigned to peers in a separate region to simulate the non-overlapping regional data. To vary data heterogeneity, we overlapped the data in each region with o percent of other regions' data. Experiments were then conducted on these different data distributions. We compared our approach with AllCascade [1], and Individual Regional Cascaded model without cross region data exchange (IRC). All these approaches were implemented in C++ and we used SVM and RSVM implementations from [11,12]. The RBF kernel and penalty cost parameters were selected using the procedure mentioned in [11] and their values are $\gamma = 2, C = 32$ for the Covertype, and $\gamma = 2^{-7}, C = 32$ for the Waveform dataset. For Satrap, the threshold value T is set to 0.75, and the cluster ratio R is set to 0.1. Results were obtained using 10-fold cross validation.

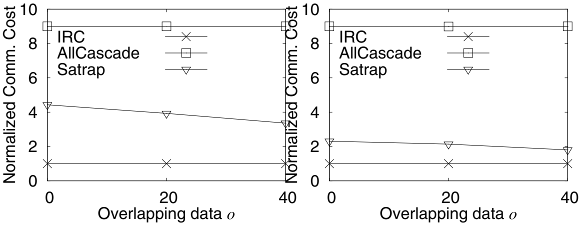
4.1 Performance Evaluation

Figures 2 and 3 present the classification accuracy (in percentage) and communication cost (as a ratio of the total dataset size in the entire network $\times 10^4$) respectively. The plots in Figure 3 are normalized to the cost of IRC which doesn't incur any inter-region costs, and are shown using a conservative 1:1 ratio between intra- and inter-region costs. This ratio can be upto 1:50 in real environments [13], so Satrap's benefits over AllCascade should be amplified.



(a) Multiclass Covertype (b) Multiclass Waveform

Fig. 2. Effect of data overlap on classification accuracy



(a) Multiclass Covertype (b) Multiclass Waveform

Fig. 3. Effect of data overlap on communication cost (normalized to that of IRC)

We varied the percentage o of overlapping data (from other regions) to simulate a varying degree of homogeneity between different regions. From Figure 2, we can see that the varying distribution does not affect the accuracy of AllCascade. However, IRC suffers as the overlap decreases. This is because IRC does not perform any data exchange between different regions, and therefore achieves reasonable accuracy only when data among different regions is homogeneous. Moreover, we observe that the Satrap achieves accuracies close to AllCascade and significantly better than IRC, with only a slight drop as the amount of overlapping data increases. However, this is accompanied by significant savings in communication cost – showing acceptable trade-off between cost and accuracy. We attribute this drop in Satrap's accuracy to the probabilistic sampling for overlapping data space (hence missing out some important data points) which is critical for saving communication cost.

By comparing Figures 2 and 3, we observe that the competing approaches are on the two extremes. IRC has the best accuracy-to-communication cost ratio, but it does not fulfil the criteria to maximize the global accuracy as it does not learn beyond the local region. Observe that the actual accuracy of IRC on average is more than 15% worse than Satrap.

On the other hand, while AllCascade has the highest accuracy, it comes with the lowest accuracy-to-communication cost ratio across all datasets. Satrap closely approximates AllCascade's accuracy while retaining a much superior accuracy-to-communication cost ratio. This ratio significantly improves as the percentage of overlapping data increases. To summarize, we observe that Satrap is able to achieve good accuracy-to-communication cost ratio in most situations.

5 Conclusion

This paper is the first effort that systematically studies the effect of network and data heterogeneity on prediction accuracy and communication cost for learning in P2P networks. Satrap, our network and data heterogeneity aware P2P classification approach, is based on a simple system of information sharing, and lends itself to easy improvement as every module can be fine-tuned depending on knowledge of the domain. Satrap achieves a better accuracy-to-communication cost ratio than existing approaches, and is justified by extensive experiments. The approach also allows users to trade off accuracy for communication cost and vice-versa. In future work, we're looking at how to mitigate the problem of low data overlap, improve the detection of data overlaps and sampling.

References

1. Ang, H.H., Gopalkrishnan, V., Hoi, S.C.H., Ng, W.K.: Cascade RSVM in peer-to-peer networks. In: ECML/PKDD, pp. 55–70 (2008)
2. Bhaduri, K., Wolff, R., Giannella, C., Kargupta, H.: Distributed decision-tree induction in peer-to-peer systems. Statistical Analysis and Data Mining 1(2), 85–103 (2008)
3. Gorodetskiy, V., Karsaev, O., Samoilov, V., Serebryakov, S.: Agent-based service-oriented intelligent P2P networks for distributed classification. In: Hybrid Information Technology, pp. 224–233 (2006)
4. Siersdorfer, S., Sizov, S.: Automatic document organization in a P2P environment. In: Lalmas, M., MacFarlane, A., Rüger, S.M., Tombros, A., Tsikrika, T., Yavlinsky, A. (eds.) ECIR 2006. LNCS, vol. 3936, pp. 265–276. Springer, Heidelberg (2006)
5. Luo, P., Xiong, H., Lü, K., Shi, Z.: Distributed classification in peer-to-peer networks. In: ACM SIGKDD, pp. 968–976 (2007)
6. Breiman, L.: Pasting small votes for classification in large databases and on-line. Machine Learning 36(1-2), 85–103 (1999)
7. Yang, B., Garcia-Molina, H.: Designing a super-peer network. In: ICDE, pp. 49–60 (2003)
8. Shin, H., Cho, S.: Invariance of neighborhood relation under input space to feature space mapping. Pattern Recognition Letters 26(6), 707–718 (2005)
9. Graf, H.P., Cosatto, E., Bottou, L., Dourdanovic, I., Vapnik, V.: Parallel support vector machines: The cascade SVM. In: NIPS, pp. 521–528 (2004)
10. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
11. 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`
12. Lin, K., Lin, C.: A study on reduced support vector machines. IEEE Transactions on Neural Networks 14(6), 1449–1459 (2003)
13. Touch, J., Heidemann, J., Obraczka, K.: Analysis of HTTP performance. Research Report 98-463, USC/Information Sciences Institute (August 1998)