P2P RVM for Distributed Classification

Muhammad Umer Khan[1], Alexandros Nanopoulos[2] and Lars Schmidt-Thieme[1]

[1] Information Systems and Machine Learning Lab, University of Hildesheim, Germany {khan, schmidt-thieme}@ismll.uni-hildesheim.de
[2] University of Eichstätt, Ingolstadt Germany alexandros.nanopoulos@ku.de

Abstract. In recent years there is an increasing interest for analytical methods that learn patterns over large-scale data distributed over Peer-to-Peer (P2P) networks and support applications. Mining patterns in such distributed and dynamic environment is a challenging task, because centralization of data is not feasible. In this paper, we have proposed a distributed classification technique based on Relevance Vector Machines (RVM) and local model exchange among neighboring peers in a P2P network. In such networks, the evaluation criteria for an efficient distributed classification algorithm is based on the size of resulting local models (communication efficiency) and their prediction accuracy. RVM, utilizes dramatically fewer kernel functions than a state-of-the-art 'support vector machine' (SVM), while demonstrating comparable generalization performance. This makes RVM a suitable choice to learn compact and accurate local models at each peer in a P2P network. Our model propagation approach, exchange resulting models with peers in a local neighborhood to produce more accurate network wide global model, while keeping the communication cost low throughout the network. Through extensive experimental evaluations, we demonstrate that by using more relevant and compact models, our approach outperforms the baseline model propagation approaches in terms of accuracy and communication cost.

1 Introduction

In recent years there is an increasing interest for analytical methods that learn patterns over large-scale data distributed over Peer-to-Peer (P2P) networks and support applications. For example, distributed classification of large amount of tagged text and image data stored in online newspapers, digital libraries and blogs. P2P matchmaking analyzes user profiles to recommend more appropriate profiles to connect with. Clustering content with respect to user's interest in media sharing P2P networks (e.g. BitTorrent, Shareaza, LimeWire etc). Other applications include collaborative and distributed spam classification (Caruana et al.(2012)) and outlier detection and scene segmentation in sensor networks.

1.1 Motivation

Mining patterns from such large-scale distributed P2P networks, is a challenging task, because centralization of data is not feasible due to prohibitive communication cost and user's privacy concerns. In P2P networks, computing devices might be connected to the network temporarily, communication is unreliable and perhaps with limited bandwidth, resources of data and computation can be distributed sparsely, and the data collections are evolving dynamically. A scheme which centralizes the data stored all over a P2P network is not feasible, because any change must be reported to the central peer, since it might very well alter the result. Therefore, the goal is to develop distributed mining algorithms that are communication efficient, scalable, asynchronous, and robust to peer dynamism, which achieve accuracy as close as possible to centralized but in-feasible ones. Recently, researchers have proposed model propagation approaches based on SVM (Papapetrou et al. (2011)) and its variants (Hock et al.(2008)) which tend to reduce model size by random sub-sampling techniques, for distributed classification in P2P networks. An inherent problem with these approaches is that, being based on SVM based classifiers and their variance, the size of resulting model (number of support vectors) typically grows linearly with the size of the training set. Therefore, such schemes incur a high communication cost required to exchange models among peers, a fact that negatively impacts the efficiency of the distributed data mining approach.

1.2 Contribution

In this paper, we have presented a distributed classification approach (P2P-RVM) for P2P networks, which is based on Relevance Vector Machines (RVM)(Tipping (2001)) and model exchange between peers with in a local neighborhood. The key feature of RVM is that while capable of generalization performance equivalent to a regular SVM, it utilizes significantly fewer kernel functions. This sparsity is achieved since posterior distributions of many of the kernel weights tend to get *zero*, during learning process. This makes it extremely effective to keep only those kernel functions which are more prototypical or relevant vectors of the local data, for making accurate predictions using compact models. Each peer in the P2P network learns an RVM model locally, and exchange this model in a synchronized way with its directly connected neighbors in the local neighborhood of network.

To perform extensive experimental evaluation of the proposed method, we have developed a simulation test-bed and compared our approach to baseline methods, which deploy variants of SVM based on random sub-sampling for model propagation in P2P networks. Experimental results demonstrate that P2P-RVM exhibits high classification accuracy and significantly reduces communication cost outperforming the bench-marked methods, and comparable in accuracy to any *state-of-the-art* centralized classifier. We also show that the proposed method is scalable i.e. independent of the size of P2P network.

The rest of this paper is organized as follows: Section 2 describes related work. Section 3 introduces our proposed distributed classification approach. Our simulation framework and experimental evaluations are described in Section 4, and the last section concludes the paper.

2 Related Work

Current *state-of-the-art* research in P2P data mining focus on developing local classification or clustering algorithms which in-turn make use of primitive operations such as distributed averaging, majority voting and other aggregate functions. Most representative work in this regard is distributed association rule mining by R.Wolff et al.(2004), distributed decision tree induction by Bhaduri et al.(2008), distributed K-Means clustering by Datta et al.(2009) and distributed classification by Ping Luo et al.(2007). Most of these locally synchronized algorithms are reactive in a sense that they tend to create a consensus in the local neighborhood of peers, by monitoring every single change in data and keeping track of data statistics, which also require extra polling messages for coordination.

Based on model propagation, an important work for distributed classification in P2P networks is by Hock et al.(2008), in which they build an RSVM (Lee and Mangasarian (2001)) model using each peer's local data, then iteratively propagate and merge the models to create an improved model. Papapetrou et al.(2011) uses SVM model exchange for this purpose. These approaches tend to rely on random sub-sampling of local data, to control the size of resulting model and then optimizing it for reduced errors. Since with SVM based classifiers, size of model grows linearly with the size of local data and many redundant support vectors also get through, these methods incur high communication cost without any significant gain in classification accuracy.

In our study, we focus on the aforementioned category of classification in P2P networks based on model propagation. Our approach is based on the idea that instead of learning models from random perturbations of data, consider the significance of each instance in data, and keep only those which are most prototypical or relevant to local data set. By using RVM as a base-learner in our model propagation, we intend to optimize both size (communication cost)and quality (accuracy) of the resulting model.

3 Approach

In this section, we present our proposed method P2P-RVM illustrating learning base classifiers locally at each peer, and iterative model propagation and update by the peers in the local neighborhood. More generally, P2P-RVM creates a 'cascade' of base-classifiers i.e. instead of analyzing whole data in one optimization step, the data is partitioned into subsets and optimized separately with multiple base-classifiers. The partial models are combined and re-learned iteratively, until the globally optimal model is obtained. We adapt this general approach for classification in a P2P network. We learn *light weight* local models on presumably naturally distributed data sets, iteratively propagate models to or receive from neighbors and update (relearn) to obtain more accurate global models.

3.1 Building Local Classifier

Classification first builds a model (denoted as classifier) based on labeled training data and then predicts class labels for new (unseen) data instances. In P2P networks, each peer contains its own training data set that is not directly available to the rest of peers. More formally, we consider an ad-hoc P2P network comprising of a set of such k autonomous peers $P = \{1, 2, \ldots, k\}$. The topology of the P2P network is represented by a (connected) graph $G(P, E)$, in which each peer $p \in P$ is represented by a vertex and an edge $\{p, q\} \in E$, where $E \subseteq \{\{p, q\} : p, q \in P\}$, whenever peer p is connected to peer q. The local training data set on a peer p is denoted as $X_p \subseteq \mathbb{R}^d$, where d is the number of data features. Finally, with $\mathcal{X} = \bigcup_{p=1}^{k} X_p$ we denote the global training data set of the entire P2P network. Please notice that $\mathcal{X}$ is not feasible to be centralized (i.e., be collected in a single peer).

Based on the local training data set X_p, each peer p can first build its local classification model m_p. However, when X_p is small, and thus not representative, the accuracy of the local model m_p is reduced. To overcome this problem, a possible solution is to learn models in a collaborative fashion, where each peer p shares its local model m_p with its immediate neighbors.

Since propagating classification models in large scale ad-hoc P2P networks, results in prohibitive communication cost, therefore, it is required to build models that are both accurate and compact, i.e. they can be represented with the *least*, as well as the most *prototypical* information, needed to be exchanged between neighboring peers.

Based on these requirements, we employ RVM, a probabilistic kernel model based on the theory of sparse Bayesian learning. The key feature of this approach is that it utilizes significantly fewer kernel functions while offering good generalization performance. This is because, inferred models are exceedingly sparse in that posterior distributions of majority of kernel weights are found to have maximum values around *zero*. Training instances associated with remaining very few non-zero weights are termed as *relevant vectors*. Below we briefly describe RVM formulation derived from Tipping(2001).

Relevance Vector Machines - Formulation:

At each peer p, given is a training set of instance-label pairs $\{(x_j, y_j)\}_{j=1}^{|X_p|}$, where $x_j \in \mathbb{R}^d$ is an input vector and $y_j \in \{-1, 1\}$ is the corresponding class label. Since we denote the size of local training data at peer p as $|X_p|$, in the following notation $|X_p|$ denotes total number of instances. Aim of classification task is to predict the posterior probability of class membership of x_j. Considering a generalized additive model $y(x)$ having the similar form as that of SVM prediction function,

$$y(x) = \sum_{i=1}^{|X_p|} w_i K\left(x, x_i\right) + w_0 \tag{1}$$

where w_i are model weights and $K(x, x_i)$ is a kernel function. Applying sigmoid logistic function $\sigma\left(y(x)\right) = 1/1 + e^{-y(x)}$, we can write the likelihood function for Bernoulli distribution of $P(y|x)$ as:

$$P\left(\hat{\mathbf{y}}|\hat{\mathbf{w}}\right) = \prod_{i=1}^{|X_p|} \sigma\left(y(x_i)\right)^{y_i} \left[1 - \sigma\left(y(x_i)\right)\right]^{1-y_i} \tag{2}$$

RVM uses the basis function $\phi(x_i) \equiv K(x, x_i)$ based on the kernel function in (1). Using this, we can re-write (2) as:

$$P\left(\hat{\mathbf{y}}|\hat{\mathbf{w}}\right) = \prod_{i=1}^{|X_p|} \sigma\left(\phi(x_i) w_i\right)^{y_i} \left[1 - \sigma\left(\phi(x_i) w_i\right)\right]^{1-y_i} \tag{3}$$

where $\hat{\mathbf{y}} = \left(y_1, \ldots, y_{|X_p|}\right)^T$, $\hat{\mathbf{w}} = \left(w_1, w_2, \ldots, w_{|X_p|}\right)^T$, and $\phi(x_i) = [1, K(x_i, x_1), K(x_i, x_2), \ldots, K(x_i, x_{|X_p|})]^T$.

RVM utilizes an $[|X_p| \times |X_p| - 1]$ basis matrix $\Phi = \left[\phi(x_1), \phi(x_2), \ldots, \phi(x_{|X_p|})\right]$, for filtering out the most relevant basis vectors, as described next.

Estimating maximum likelihood for (3), with as many parameters w as the training examples, would lead to severe over-fitting. To avoid this, RVM puts a constrain on parameters by explicitly defining the following *prior* probability distribution over them, using the principle of *automatic relevance determination (ARD)* proposed by MacKay (2004).

$$p(\hat{\mathbf{w}}|\hat{\alpha}) = \prod_{i=0}^{|X_p|} \mathcal{N}\left(w_i|0, \alpha_i^{-1}\right) \tag{4}$$

with α a vector of $|X_p{+}1|$ hyper-parameters i.e. each hyper-parameter moderates the strength of the weight with which it is associated. Hyper-parameters α are estimated from the training data using Gamma distribution with uniform scales.

$$p(\hat{\alpha}) = \prod_{i=0}^{|X_p|} Gamma\left(\alpha_i|a, b\right) \tag{5}$$

Using a broad prior over the hyper-parameters α, posterior probability of the associated weights approaches to zero, thus considering those inputs as *irrelevant*. This key feature of RVM is ultimately responsible for significantly reducing the number of basis functions (and corresponding support vectors) to most *relevant* ones. These support vectors are considered to be the most *prototypical* representatives of data set. Ultimately, RVM maximizes the posterior probability of class labels parameterized by hyperparameters α, which is known as maximizing *marginal likelihood*.

For detailed explanation of inference procedure and hyper-parameter optimization, we refer the reader to original paper by Tipping (2001), for reasons of space limitation.

3.2 Model Exchange and Update

Based on the local training data set X_p, each peer p can first build its local classification model m_p, as described in previous section. Let N_p denote the set of immediate neighbors of peer p, i.e., $N_p = \{q \in P | q \neq p, \{p, q\} \in E\}$. After learning the local model, a peer p uses its neighbor list N_p to propagate m_p to all directly connected neighbors. Moreover for receiving models, each peer p waits for time t until m_q from all the $q \in N_p$ have been received. Once all the neighboring models have been received, each peer updates its local model with the support vectors in the received ones. The resulting global model built through this collaborative process is more accurate and helps improving the classification performance of the whole P2P network. Algorithm 1 describes the working of P2P-RVM at a local peer p.

Algorithm 1 P2P-RVM algorithm for peer p

```
Input: X_p = Local training data set, t = Time to wait for receiving models before
    updating, N_p = List of neighbors
Output: Updated model M
  Train local classifier model m_p using RVM on X_p
  foreach q ∈ N_p do
    Propagate the support vectors of m_p to q   ▷ Exchange with direct neighbors
  end for
  RECEIVED_p := ∅              ▷ Initialize an empty set to keep received models
  while waiting_time < t do
    if receive_request then              ▷ Handle receive requests from neighbors
      if m_q ∉ RECEIVED_p then
        Send ACK
        RECEIVED_p := RECEIVED_p ∪ {m_q}
      end if
    end if
  end while
  if RECEIVED_p ≠ ∅ then
    foreach m_q ∈ RECEIVED_p do                              ▷ Merge all models
      m_p = m_p ∪ m_q
    end for
  end if
  M = RVM model trained using updated m_p
  return M
```

Since P2P networks are highly dynamic i.e. peers usually leave and join the network in an ad-hoc manner. Model propagation approach implicitly deals with such *peer dynamism*, because even if a peer leaves the network, its local knowledge remains in the network in the form of its model, it had shared with other peers. Moreover, as new data keeps arriving in a P2P network, our simulation consider this data as a new peer, and executes Algorithm-1 for it, consequently dealing with *data dynamism.*

4 Experiments and Results

In this section, we present our simulation setup for P2P network, experiments with P2P-RVM and the baseline methods and finally the evaluations to compare their performance. We have performed evaluations based on two most significant criteria for the problem of learning in P2P networks i.e. classification accuracy and communication cost. The communication cost is measured as the sum of size of all propagated models, whereas the size of each model is measured as the number of support vectors it contains. We have compared P2P-RVM with state-of-the-art model propagation technique for distributed classification called Cascade Reduced-SVM proposed by Hock et al.2008. The performance of two methods is also compared with standard SVM, especially to analyze how better they perform relative to any state-of-the-art centralized classifier. Finally, to demonstrate the effectiveness of model exchange, we also consider a baseline that performs classification only locally, without any model exchange.

4.1 Experimental Setup

Our evaluation needs to determine the network topology with edge delays and local computations at each peer with message exchange. For this purpose we used the BRITE topology generator for P2P networks, with ASWaxman model. Other BRITE parameters we used are $HS = 1,000$, $LS = 100$ (size of plane) and constant bandwidth distribution with $MaxBW = 1,024$ and $MinBW = 10$ (please refer to BRITE documentation for more details: www.cs.bu.edu/brite).

For learning regular SVM as a centralized baseline, we used C-SVC implementation provided by LibSVM (Chih-Chung et al. 2011). We used RBF kernel for learning RVM, RSVM and SVM classifiers. Optimal values of hyperparameters such as kernel width for RVM, C and γ for RSVM and SVM, were found using 10-fold cross validation.

4.2 Data Sets

We have used two standard benchmark classification data sets for our experiments. These are,

- *covertype* (581012×54, 7 classes) data set from UCI repository, and
- *cod-rna*(488565×8, 2 classes) data set from LIBSVM repository.

Both data sets are among the largest in these widely used repositories. In recent literature of P2P classification, *covertype* data has been used by several researchers to evaluate their models. Data from above mentioned repositories, is already partitioned into training and test sets. Data in both training and test sets, is distributed uniformly among the peers of the network, before performing classification task.

4.3 Results

Algorithms were compared with respect to quality/cost ratio i.e. what accuracy can be achieved with a given communication cost. Communication cost is given relative to the upper bound of central scheme, which is cost of centralizing the network's whole data to some server. Figure 1 illustrates average prediction performance in correlation to communication cost for the whole network, for a complete execution of each algorithm on a network of 50 peers.

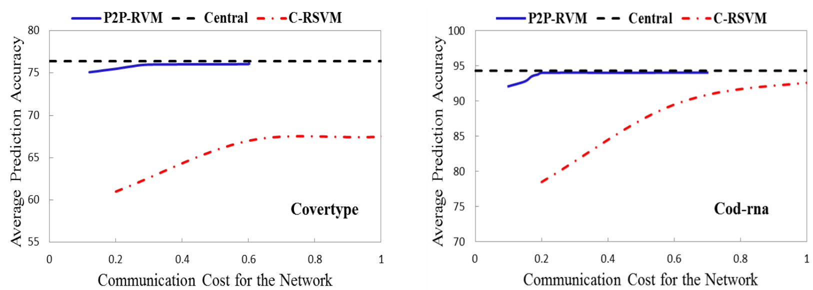

Fig. 1: Classification quality and communication cost

The results clearly show that P2P-RVM out-performs the baseline Cascade-RSVM (denoted as C-RSVM) by achieving average accuracy which is quite close to that of centralized SVM (denoted as Central), by utilizing only 20 percent of the communication cost of C-RSVM and in-feasible Central approach. Table-1 compares the Central, C-RSVM and P2P-RVM in terms of average accuracy and the later two in terms of average number of support vectors (nSV) used per peer, for a network of 50 peers.

Table 1: **Comparison of average accuracy for whole network and average model size per peer (nSV)**

Data Sets	Accuracy (%)			Average nSV per peer	
	Central	C-RSVM	P2P-RVM	C-RSVM	P2P-RVM
covertype	76.5	69.5	75.9	174±0	70±5
cod-rna	94.3	93.4	94.1	195±0	10±1

Figure 2 shows the performance of P2P-RVM in terms of scalability i.e. influence of network size (no. of peers). Secondly, it also illustrates the effectiveness of model propagation approach by comparing it's accuracy with that of local models (denoted by L-RVM and L-RSVM) i.e. models learned without any exchange.

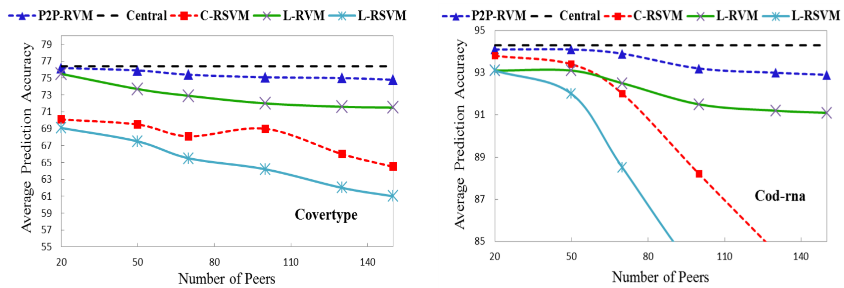

Fig. 2: Scalability: Effect of number of peers in the network on average accuracy

Model exchange for collaborative classification significantly improves the network wide prediction accuracy. Moreover, model exchange seems to get more beneficial, as the network size increases (especially in case of P2P networks, there is a majority of free riders with very little amount of data to perform any meaningful classification). Figure 2 also depicts the scalability of P2P-RVM as compared to C-RSVM. P2P-RVM has shown high resilience to performance degradation as the network size increases. On the other hand, C-RSVM exhibits a decline in accuracy, as local data sets get smaller in size.The reason for this performance is that, RSVM algorithm significantly depends on the size of data as it uses a random subset (n_u percent of data) to be considered as support vectors while learning the model. Whereas, RVM approach considers only the most prototypical vectors following the principal of automatic relevance detection.

5 Conclusion

In this paper, we proposed P2P-RVM, a collaborative distributed classification approach which utilizes relevance vector machines to learn local models, and exchange them among peers in the local neighborhood of a P2P network through cascade model propagation and updates. P2P-RVM has shown a strong performance benefits in terms of classification accuracy, communication cost and scalability.

In our future work we will investigate the problem of classification in dynamic distributed networks, where nodes are allowed to physically move in space, such as in case of vehicular ad hoc networks (VANET). We expect that, approaches like RVM, can be useful in scenarios where communication resources are scarce but distributed learning applications still need to be highly accurate.

Acknowledgment

This work is funded by the Seventh Framework Program of European Commission, through the project REDUCTION (No. 288254). `www.reduction-project.eu`.

References

Bhaduri, Ran Wolff, Chris Giannella, and Hillol Kargupta. (2008): Distributed Decision-Tree Induction in Peer-to-Peer Systems. Stat. Anal. Data Min. 1, 2, 85-103.

Byung-Hoon Park, Hillol Kargupta. (2002): Distributed Data Mining: Algorithms, Systems, and Applications. In: The handbook of data mining, pp. 341-358 edited by Nong Ye

Chih-Chung Chang and Chih-Jen Lin, LIBSVM (2011): A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011.

Caruana, Godwin and Li Maozhen,(2012):A survey of emerging approaches to spam filtering. In: *ACM Computing Surveys*

Datta, Bhaduri, Giannella, Wolff, Kargupta (2006): Distributed Data Mining in Peer-to-Peer Networks. Internet Computing, IEEE , vol.10, no.4, pp.18,26, July-Aug. 2006

Datta, Chris Giannella, and Hillol Kargupta. (2009): Approximate Distributed K-Means Clustering over a Peer-to-Peer Network. IEEE Trans. on Knowl. and Data Eng. 21, 10, 1372-1388

Graf, Cosatto, Bottou, Dourdanovic,Vapnik.(2004): Parallel support vector machines: The cascade SVM. In Advances in neural information processing systems (pp. 521-528)

Hock-Hee Ang, Vivekanand Gopalkrishnan, Steven C. Hoi, and Wee Keong Ng. (2008). Cascade RSVM in Peer-to-Peer Networks. In: European Conference on Machine Learning and Knowledge Discovery in Databases.

Lee, Y. and Mangasarian, Olvi L.(2001): RSVM: Reduced Support Vector Machines. *First SIAM International Conference on Data Mining,5-7*

MacKay, D. J. (1996). Bayesian methods for back propagation networks. In Models of neural networks III (pp. 211-254). Springer New York.

Papapetrou Odysseas , Wolf Siberski, and Stefan Siersdorfer. (2011): Collaborative classification over P2P networks. In: 20th international conference companion on World wide web (WWW '11)

Ping Luo, Hui Xiong, Kevin L, and Zhongzhi Shi.(2007): Distributed classification in peer-to-peer networks. In: 13th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'07)

R. Wolff and A. Schuster. (2004): Association rule mining in peer-to-peer systems. In: Trans. Sys. Man Cyber. Part B 34,6.

Sangkyun Lee, Marco Stolpe, and Katharina Morik. (2012): Separable approximate optimization of support vector machines for distributed sensing. In:European conference on Machine Learning and Knowledge Discovery in Databases - Volume Part II (ECML PKDD'12)

Tipping, Michael E.(2001): Sparse Bayesian Learning and the Relevance Vector Machine. *Journal of Machine Learning Research,211-244.*

Yumao Lu, Roychowdhury, Vandenberghe. (2008): Distributed Parallel Support Vector Machines in Strongly Connected Networks. In: IEEE Transactions on Neural Networks, vol.19, no.7, pp.1167,1178.

Zeng, Li, Ling Li, Lian Duan, Kevin Lu, Zhongzhi Shi, Maoguang Wang, Wenjuan Wu, and Ping Luo.(2012): Distributed data mining: a survey. Information Technology and Management 13, no. 4: 403-409.