



Journal of Applied Sciences

ISSN 1812-5654

science
alert

ANSI*net*
an open access publisher
<http://ansinet.com>

A Survey of Distributed Classification Based Ensemble Data Mining Methods

D. Mokeddem and H. Belbachir

Laboratory of Signal, Systems and Databases LSSD, Department of Computer Sciences,
University of Sciences and Technology Mohamed Boudiaf, B.P. 1505, El Mnaouer, Oran, Algeria

Abstract: Distributed classification is one task of distributed data mining which allows predicting if a data instance is member of a predefined class. It can be applied for two different objectives: the first is the desire to scale up algorithms to large data sets where the data are distributed in order to increase the overall efficiency; the second is the processing of data which are inherently distributed and autonomous. Ensemble learning methods as very promising techniques in terms of accuracy and also providing a distributed aspect, can be adapted to the distributed data mining. This study presents a survey of various approaches which use ensemble learning methods in a context of distributed classification, using as base classifier decision trees algorithm. According to the two objective mentioned above, the majority of work reported in the literature address the problem using one of the two techniques. The adaptation of ensemble learning methods to disjoint data sets, in the context of mining inherently distributed data and the parallelization of ensemble learning methods, in a scalability context. Through this survey, one can deduct that the work done in one or the other perspective (scaling up data mining algorithms or mining inherently distributed data) could be complementary. Some open questions, current debates and future directions are also discussed.

Key words: Distributed data mining, ensemble learning methods, decision trees algorithm

INTRODUCTION

Distributed data mining is the classic data mining process of extracting non-trivial information from distributed data sources, with a minimum of interaction between data sites (Park and Kargupta, 2002), it is a research area which caused many interests in the literature these last years (Liu *et al.*, 2006). In addition to this distributed aspect that it is necessary to take into account, this approach has also inherited from the classical data mining two major challenges: increase accuracy and decrease computing time (Aounallah and Guy, 2007).

This process has been used in two different contexts. The first is mining inherently distributed data where data must be processed in their local sites because of several constraints such as the storage and computing costs, communication overhead and privacy. The second context is scaling up used algorithms; in this case, data set can be partitioned and distributed through different sites and then data mining process is applied simultaneously on smaller data subsets.

Ensemble learning methods (a.k.a ensemble data mining methods) are one of the most active areas of research in supervised learning. The basic idea is to

combine the results of several data mining algorithms, each one applied to a diversified data set. This approach has been often much more accurate than the classic individual approaches.

At first sight, this strategy of working group could correspond to the constraints of the distributed data mining, but several tracks still remain to be explored. Indeed, in the first study related to ensemble learning, the algorithms are based on the same whole data set perturbed by ad hoc methods. In distributed data mining, data subsets represent generally horizontal partitioning of the whole data set. Several adaptations are then necessary to benefit from ensemble learning methods in distributed data mining.

The classification has been recognized as one of the core tasks in data mining and also in distributed data mining. It can be performed by several methods. This survey describes a state of the art of decision trees techniques in ensemble learning methods and distributed data mining, emphasizing recent developments and focusing more on distributed classification using ensemble learning methods.

The aim of the present contribution is to present the main techniques in distributed ensemble learning methods by grouping them into two categories:

- Techniques based predicting from distributed data approach (Sabine McConnel and David Skillicorn, 2004; Skillicorn and McConnell, 2008)
- Techniques based scalling-up ensemble learning methods approach, which is also divided in two subcategories:
 - Methods using disjoints data sets (Chawla *et al.*, 2001, 2004; Kamath and Cantú-Paz, 2001)
 - Methods using parallelism (Yu and Skillicorn, 2001; Lazarevic and Obradovic, 2002; Dai *et al.*, 2005)

DISTRIBUTED DATA MINING

From the literature related to the distributed data mining, one can locate two techniques used. The first are parallel techniques which often calls upon dedicated machines and parallel processes communication tools. The second are aggregation techniques (Aounallah and Guy, 2007) that aggregate data, predictions, or basic models.

Parallel techniques: These techniques are based on extracting parallel aspect from used algorithm, thus allowing parallel execution of several parts of the algorithm on several processors. This requires dedicated architectures such as shared memory systems. Grid computing represents currently an interesting alternative as plate form for parallel computing (Tsai and Chao-Tung, 2004).

Parallelization of data mining algorithms can be performed according to two approaches: data parallelism or task parallelism (Park and Kargupta, 2002). In practice, these techniques require a particular intention on several factors such as: load balancing and the cost of communication necessary for data exchange between processors.

Several studies has exploited the divide and conquer nature of decision trees techniques and have proposed parallel techniques such as Parallel SLIQ (Mehta *et al.*, 1996), parallel SPRINT (Shafer *et al.*, 1996) and RainForest (Gehrke *et al.*, 1998). These systems attempt to optimize the processing time by the use of specific data structures and parallelism.

Aggregation techniques: These techniques are used in order to mining data from multiple distributed databases. They can be classified according to whether the data mining algorithm uses a fraction of the available data set; in this case we speak of data aggregation, or all data, which is either model aggregation or prediction aggregation (Aounallah and Guy, 2007).

Data aggregation techniques: Data mining algorithm uses samples from several databases. These samples can be data subsets chosen from databases (Davies and Edwards, 2000), or a few well-chosen databases (Zhang *et al.*, 2003). The disadvantage of data aggregation is the transfer time as well as the processing time for selecting the right samples.

Model aggregation techniques: They consist in aggregating the models which are constructed individually from distributed databases. In classification for example, this technique combines classification rules to produce a single whole of rules (Lawrence Hall *et al.*, 1998, 2000; Bradley *et al.*, 2002).

Prediction aggregation techniques: They build an ensemble of base classifiers of which the predictions will be combined in order to classify new data whose class is unknown. These techniques commonly called ensemble learning were conceived initially to one whole data set in order to improve prediction accuracy.

Although, these techniques produce only predictive models which cannot explain the choice of their classifications to an analyst (Aounallah and Guy, 2007), they present a promising alternative for distributed data mining. This is due to the following factors:

- Their contribution in increasing accuracy
- Distributed aspect that they offer
- The elimination of global classifier construction phase, as in the case of model aggregation, could accelerate the data mining process

ENSEMBLE LEARNING METHODS IN CLASSIFICATION

Classification is one task of data mining which allows predicting if a data instance is member of a predefined class. Input is a training data set S , where each instance is typically represented in the form of vector attributes $x = \langle x_1, x_2, \dots, x_m \rangle$, y is the class attribute. The objective of classification is to train a classification algorithm A on training data set S to find a good approximation of a certain function $f(x) = y$. The approximate function Cl is called classifier. Evaluation of Cl accuracy is performed with a data set T independent of S . The classifier will be thereafter able to predict the class value y for new data d by calculating $Cl(d)$.

In the case of ensemble learning methods, N base classifiers Cl_i are constructed from N data sets S_i . The classification of new data is made by combining predictions of the N base classifiers using typically a

majority vote. In spite of the simplicity of this intuitive idea union makes the force, it is based on a statistical theory reinforced by several empirical studies. These studies showed in various researches (Breiman, 1996, 2001; Freund, 1995; Kim, 2002; Ross Quinlan, 1996) that the accuracy of a learning algorithm can be improved in a significant way by applying perturbing and combining methods.

The basic principle is to generate multiple versions of the classifier by perturbing the training set, construction method or some parameters (Robert Duin, 2002). The most appropriate algorithms to the application of this approach are those considered as unstable, i.e., small perturbations in their training sets or in construction may result in large changes in the constructed classifier. The experimental results show that 50 base classifiers are in general sufficient (Breiman, 1996), but the processing time is still a field of investigation.

In the following, the most popular approaches in training set generation and combination techniques will be presented. Recent works based on decision trees method will be also presented.

Generation of ensembles learning: In order to generate multiple versions of training data set, several techniques can be applied, among the most used, we present the techniques: bagging, boosting, cross validated committees and random subspace method.

Bagging (Breiman, 1996) (Bootstrap AGGREGatING): This method perturbs the training set repeatedly to generate multiple base classifiers and combines these by simple voting (cf. 3.2). A sampling method with replacement (bootstrap) is applied N times from the training data set S (containing N instances), forming the resampled training set S' , who is of the same size than S . Some instances in S may not appear in S' , some may appear more than once.

Boosting (Robert Shapire, 1990): This method works with weights of training examples. A sequence of classifiers is created in respect to modifications of the training set. For each next iteration, the weights of training examples, which were classified incorrectly by the previous classifier, are increased. The weights of those training examples, which were classified correctly, are decreased. In this way, the learning of next classifier focuses on incorrectly classified training examples. The prediction of the resulting classifier is given as a weighted combination of individual classifier predictions. Adaboost (Freund and Robert Schapire, 1995) is a typical algorithm that uses this technique.

Cross validated committees (Dietterich and Thomas, 2000): This technique consists in dividing the training data set S into K disjoint data subsets $\{S_1, S_2, \dots, S_K\}$. The following process is repeated K time: construction of a classifier Cl_i with S private of S_i ($S-S_i$) then to evaluate the accuracy of Cl_i tested out of S_i . The total accuracy is obtained by the average. The data sets built in this manner are called cross-country race validated committees.

Random subspace method (Kam Ho, 1998): This technique consists in selecting a certain number of attributes from the original attribute space; training data subsets obtained will be used to construct the base classifiers. It is an approach that is very beneficial to the problems with a large number of attributes, with multiple redundancies.

Combination methods: Once the base classifiers built, various techniques can be used to combine the results of each classifier. The most cited in the literature are: the majority vote, the weighted vote and stacking.

The majority vote: It is a simple and intuitive technique, which is to classify the new instances according to majority prediction of the base classifiers. The disadvantage of this method is in the case where more than half of base classifiers get false results.

The weighted vote: It is a vote based on weights associated with classifiers. These weights may be reduced or increased as the train classifiers; according to they produce a good or a bad prediction.

Stacking: It is a method that combines several base classifiers. The first phase is to induce N classifiers Cl_i , from N data sets $\{S_1, S_2, \dots, S_N\}$. The evaluation is then made on a test data set $T = \{t_1, t_2, \dots, t_j\}$, independent of training data sets S_i . In the second phase, a new training data set M is formed by computed values $Cl_i(t_j)$ and the true class of the instance t_j , $class(t_j)$. Each instance of is of the form $\langle Cl_1(t_j), Cl_2(t_j), \dots, Cl_N(t_j), class(t_j) \rangle$. In the last step, a global classifier is constructed from data set M . The base classifiers can be constructed by various algorithms (decision trees, neural networks...) depending on the problem (David Wolpert, 1992).

Decision trees as base classifier: Decision trees are widely used in classification because they have decent accuracy and moreover are easier to interpret. Their nature unstable also makes them good candidates for ensemble learning methods. Indeed, a lot of study in Bagging and

Table 1: Decision trees in ensemble learning methods: some related studies

Research works	Methods of creating diversity	Method of combining results
RandomForest (Breiman, 2001)	Bagging	Majority vote
	Random subspace method	Weighted vote
CascadingTrees (Li and Huiqing, 2003)	K decision trees are built from the first k attributes, sorted by information gain	Weighted vote based on the concept of coverage
RotationForest (Rodriguez <i>et al.</i> , 2006)	Attribute space is randomly devised in k data subsets and principal component analysis method is performed in data subsets	Majority vote

Boosting techniques are based on decision trees algorithms, such as C4.5 algorithm (Ross Quinlan, 1993). Significant improvements in C4.5 accuracy have been obtained in many works by applying ensemble learning approach (Dietterich and Thomas, 2000). Among the latest uses, we present an overview of the algorithms RandomForest, CaScading Tees and RotationForest (Table 1).

RandomForest (Breiman, 2001): This algorithm constructs training data sets using Bagging technique. Diversity is reinforced by random subspace method. Computing information gain is optimized by a random choice of M attributes (M is a parameter algorithm). Empirical results showed a significant accuracy improvement compared to standard C4.5 and even the approaches Bagging and Boosting.

CaScading trees (CS4) (Li and Huiqing, 2003): CS4 CaScading-and-combination FOR constructing decision tree ensembles has been designed with the aim to improve accuracy of C4.5 algorithms family. Unlike the bagging and boosting that create diversity by changes in the basic data, CS4 uses the same data set to build a set of decision trees. The changes are carried out on training phase of base classifiers. K decision trees are built from the first k attributes, sorted by information gain; the *i*th attribute node will be the root of *i*th tree. This method was tested on biomedical data of high dimension (more than 10,000 attributes) and gave interesting results. CS4 also gives the opportunity to have a meaningful global model, which is impossible in conventional methods.

RotationForest (Rodriguez *et al.*, 2006): The base classifiers are decision trees built independently, but each tree is trained on the total data in an attributes space in rotation. In order to create training data set for a base classifier, attribute space is randomly devised in k data subsets and principal component analysis method is performed in data subsets.

The objective of this method is to improve individual accuracy of each base classifier and to have diversity in the ensemble. The experimental results reported in this work say that RotationForest can provide better accuracy than RandomForest, Boosting and Bagging.

ENSEMBLE LEARNING METHODS IN DISTRIBUTED DATA MINING

Many studies have used ensemble learning methods in distributed data mining context, according to the two perspectives included: either mining inherently distributed data and scaling up ensemble learning methods. A summarization of some related works is made in Table 2, according methods of data partitioning and methods of combining results.

Predicting from distributed data: In applications where data are geographically distributed and can not be mined entirety in a central site, it is legitimate to ask whether we can be satisfied with the natural diversity presents a priori in data subsets without cause perturbations as in conventional ensemble learning methods.

The study presented by Sabine McConnell and David Skillicorn (2004) studied the effect of vertical partitioning of the data set on classification performance in a distributed environment, noting that such partitioning is not much addressed in distributed data mining literature, compared with horizontal partitioning. For each partition, a decision tree is constructed and the predictions are then combined with two schemes: the majority vote and weighted voting.

A more recent study (Skillicorn and McConnell, 2008) presents a technique called ensemble attribute that allows prediction from vertically partitioned data. This technique is evaluated according several factors: the accuracy compared to a centralized prediction, the total size of decision trees and the processing time. A problem seen still open is whether the attributes that are correlated have an effect on performance when placed on the same site, or if they are separated through different sites.

Although, the outlook is different, the study of prediction from distributed data by a horizontal partitioning can draw on the work done with the aim of scaling up, including the approach of disjoint subsets, to be presented in the following paragraph.

Scaling up ensemble methods: Typically, a classical data mining algorithm is designed to handle between 200 and 100,000 training examples, in seconds or minutes, on a platform office (Blake and Merz, 1998). For data mining

Table 2: Ensemble learning methods based on decision trees in distributed data mining : some related studies

Perspective	Research works	Methods of data partitioning	Methods of combining results	
Predicting from distributed data	Sabine McConnel and David Skillicorn (2004)	Vertical partitioning	Majority vote	
		Horizontal partitioning	Weighted vote	
	Skillicorn and McConnell (2008)	Vertical partitioning	Majority vote	
		Horizontal partitioning	Weighted vote	
Scaling-up ensemble learning methods				
Disjoint data set approach	Chawla <i>et al.</i> (2001)	Random vertical data partitioning	Majority vote	
		Clustering	Weighted vote	
	Chawla <i>et al.</i> (2004)	IVote, RVote (Breiman, 1999).	Majority vote	
		Kamath and Cantú-Paz (2001)	Sampling at the nodes	Weighted vote
	Parallelism	Dai (2005)	Bagging	Majority vote
			Random subspace method	
Lazarevic and Obradovic (2002)	Yu and Skillicorn (2001)	Boosting	Majority vote	
		Random vertical data partitioning	Weighted vote	
		Random vertical data partitioning	Majority vote	

community, very large databases contains from 100,000 with a dozen of attributes. Indeed, with a data size in the range of tera-byte and hundreds of attributes to handle, scalability issue is to see if the algorithm can effectively deal with a very large data set from which we want to build the best possible models. This becomes even more complicated in the case of ensemble methods, which data set processing is repeated tens of time or more.

Scaling up ensemble methods is usually through two techniques: the data reduction technique by manipulating disjoint subsets and parallelism technique.

Disjoints data sets approach: The conventional algorithms of ensemble learning approaches construct each base classifier from data set of the same size as the original data set. The first experiments of bagging, for example, were in the context of small data sets, up to 20,000 examples (Chawla *et al.*, 2003). But in high-performance objective in distributed data mining, the size of the data set should be much more important. One simple way is to select subsets smaller, which could reduce the processing time, but it is difficult, at first, to know if the accuracy will not be affected.

Breiman presents an algorithm which allows choosing from the entire data set random subsets (Breiman, 1999). It is an approach that can handle very large data sets, but the number of examples used by each classifier is approximately 800 examples in discussed experiments, which requires the use a very large number of classifiers.

Chawla *et al.* (2001) explored different strategies for partitioning training data set. The experimental results show that a simple random data partitioning in several disjoint subsets, generates greater accuracy compared to the creation of several bags of the same size and a considerable gain in processing time. The application of a more intelligent method to data partitioning which is clustering has been even more beneficial than just random partitioning.

The diversity of classifiers is crucial in ensemble learning approach; it can also be obtained by sampling at

the nodes in the case of decision trees as in (Kamath and Cantú-Paz, 2001), where the calculations at tree nodes use only random part of training data set. In general, competitive accuracy values were found compared to Boosting and Bagging, except for very large data set.

In a more recent study (Chawla *et al.*, 2004), the base classifiers are also constructed from different data disjoints subsets using IVotes and RVotes algorithms designed by Breiman (1999). The results demonstrate the possibility of constructing hundreds of base classifiers, with data sets of very limited size. This study has obtained accuracy values similar or better than boosting, bagging and distributed boosting (cf. 4.2.2), with a considerable gain in processing time.

Parallelism: The parallel aspect is very visible through the possibility of building base classifiers simultaneously. The parallelism of Bagging is relatively more straightforward. A study presented by Yu and Skillicorn (2001) proposed partitioning data randomly and evenly, across multiple processors. Each processor executes the sequential algorithm on its local data to obtain suitable predictions.

The technique of boosting has been widely used in order to improve the accuracy of a classifier, on a single centralized data set of a size sufficiently small to fit in central memory of a typical computer. Contrary to the approach of Bagging where base classifiers are built in an independent manner, in the standard approach of Boosting the base classifiers are built in series. The training is done on the entire data set, assigning weight values to the data at each cycle.

A study of Lazarevic and Obradovic (2002) proposed a parallel version of Boosting algorithm, which can be applied in the case where disjoint data sets can not be treated as a single data base. The base classifiers are constructed in parallel from disjoint data sets. Vectors of local weights are updated at each site and are communicated to all sites. The experimental results show that the proposed algorithm has accuracy comparable or

even better than Boosting technique applied on union of distributed data sets. RandomForest algorithm also suffers from the high cost of processing time despite its important contribution in terms of accuracy. A parallel version of RandomForest is studied by Dai *et al.* (2005).

DISCUSSION

Distributed data mining come from two different but complementary objectives: the first is the desire to scale up algorithms to large data sets where the data are distributed by the algorithm in order to increase the overall efficiency; the second is processing data which are inherently distributed and autonomous.

Scalability problematic concerns conventional data mining algorithms and all the more ensemble learning methods, which process on tens of data sets, with 100% of the original size. A first straightforward solution is using parallelism technique; other solution is to reducing the size of learning ensembles. This second approach ask the question: Does all of the data set is really useful to build the best possible models? Outlying literature that deals with this subject, one can see that this has not yet reached a general consensus. Indeed, some note that the data mining reported better results if more data are analyzed (Breiman, 1999), others say that the sampling of a very large data set can simplify the learning task, it can also degrade the accuracy (Perlich *et al.*, 2003).

Many others studies hand that using a sampling operation can improve the accuracy (Eschrich *et al.*, 2002). If indeed we do not need all the data, several tracks are still to be explored: sampling methodology, adequate sample size, maintaining acceptable accuracy. On the other hand, another question arises: is to see if the reduction of data set could be done by data or attributes. The authors in (Bryll *et al.*, 2003) for example support the proposal that attribute partitioning (vertical) is better than data partitioning (horizontal) into ensemble learning methods.

In order to mining distributed data, it is legitimate to ask whether we can be satisfied with the natural diversity presents a priori in the subsets of data, without causing perturbations as in the conventional ensemble learning methods. In such applications, data partitioning is performed by data, or by attributes.

CONCLUSION

Ensemble learning methods have been formally and empirically shown to outperform single learners in many cases. In classification task, it consists in creating multiple base classifiers and then combining their predictions. This study has focused on using of ensemble learning methods in distributed data mining context, which still a

potential for study. Diversity notion used in ensemble methods inspires a large number of approaches combinations which should be studied. On the other hand, the study done in one or the other perspective (scaling up data mining algorithms or mining inherently distributed data) could be complementary.

REFERENCES

- Aounallah, M. and M. Guy, 2007. Distributed data mining: Why do more than aggregating models. Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 07), 2007, Hyderabad, India, pp: 2645-2650.
- Blake, C.L. and C.J. Merz, 1998. UCI Repository of Machine Learning Databases. 1st Edn., University of California, Irvine, CA.
- Bradley, P., G. Johannes, R. Raghu and S. Ramakrishnan, 2002. Scaling mining algorithms to large databases. *Commun. ACM*, 45: 38-43.
- Breiman, L., 1996. Bagging predictors. *Mach. Learn.*, 24: 123-140.
- Breiman, L., 1999. Pasting small votes for classification in large database and on-line. *Mach. Learn.*, 36: 85-103.
- Breiman, L., 2001. Random forest. *Mach. Learning.*, 45: 5-32.
- Bryll, R., G.O. Ricardo and O. Francis, 2003. Attribute bagging: Improving accuracy of classifier ensembles by using random feature subsets. *Pattern Recogn.*, 36: 1291-1302.
- Chawla, N.V., S. Eschrich and L.O. Hall, 2001. Creating ensembles of classifiers. Proceedings of the IEEE International Conference on Data Mining, 2001, USA., pp: 580-581.
- Chawla, N.V., T.E. Moore, L.O. Hall, K.W. Bowyer, W.P. Kegelmeyer and C. Springer, 2003. Distributed learning with bagging-like performance. *Pattern Recogn. Lett.*, 24: 455-471.
- Chawla, N.V., O. Lawrence Hall, W. Kevin Bowyer and W. Philip Kegelmeyer, 2004. Learning ensembles from bites: A scalable and accurate approach. *J. Mach. Learn. Res.*, 5: 421-445.
- Dai, I., L. Jooan and C. Morgan Wang, 2005. Efficient parallel data mining for massive datasets: Parallel random forest classifier. Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA), June 2005, CSREA Press, DBLP., pp: 1142-1148.
- David Wolpert, H., 1992. Stacked generalization. *Neural Networks*, 5: 241-259.
- Davies, W. and P. Edwards, 2000. Dagger: A new approach to combining multiple models learned from disjoint subsets. *Machine Learning*, 2000: 1-16.

- Dietterich, T.G. and G. Thomas, 2000. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization. *Mach. Learn.*, 40: 139-157.
- Eschrich, S., N.V. Chawla and L.O. Hall, 2002. Learning to predict in complex biological domains. *J. Syst. Simul.*, 2: 1464-1471.
- Freund, Y. and E. Robert Schapire, 1995. A decision-theoretic generalization of on-line learning and an application to boosting. *Proceedings of the Second European Conference on Computational Learning Theory, 1995, LNCS.*, pp: 23-37.
- Freund, Y., 1995. Boosting a weak learning algorithm by majority. *Inform. Comput.*, 121: 256-258.
- Gehrke, J., R. Ramakrishnan and G. Venkatesh, 1998. Rainforest – a framework for fast decision tree construction of large datasets. *Proceedings of the 24th International Conference on Very Large Data Bases, 1998, New York, NY, USA.*, pp: 416-427.
- Kam Ho, T., 1998. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20: 832-844.
- Kamath, C. and E. Cantú-Paz, 2001. Creating ensembles of decision trees through sampling. *Proceedings of the 33-rd Symposium on the Interface of Computing Science and Statistics, 2001, Costa Mesa, CA.*, pp: 1-1.
- Kim, Y., 2002. Convex hull ensemble machine. *Proceedings of IEEE International Conference on Data Mining (ICDM), 2002, IEEE Computer Society, Maebashi City, Japan*, pp: 243-249.
- Lawrence Hall, O., C. Nitesh and W. Kevin Bowyer, 1998. Combining decision trees learned in parallel. *Proceedings of the Distributed Data Mining Workshop at International Conference of Knowledge Discovery, 1998, Department of Computer Science and Engineering*, pp: 1-6.
- Lawrence Hall, O., W. Kevin Bowyer, W. Philip Kegelmeyer, E. Thomas Moore and C. Chi-ming, 2000. Distributed learning on very large data sets. *Proceedings of the Workshop on Distributed and Parallel Knowledge Discovery, 2000, KDD00*, pp: 79-84.
- Lazarevic, A. and Z. Obradovic, 2002. Boosting algorithms for parallel and distributed learning. *Parallel Distributed Data Mining*, 11: 203-229.
- Li, J. and L. Huiqing, 2003. Ensembles of cascading trees. *Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03), 2003, Melbourne, Florida, USA.*, pp: 1-4.
- Liu, K., H. Kargupta and J. Ryan, 2006. Distributed data mining bibliography. *Release*, pp: 1-7. <http://www.biostat.wustl.edu/archives/html/s-news/2003-09/msg00165.html>.
- Mehta, M., R. Agrawal and J. Rissanen, 1996. SLIQ: A fast scalable classifier for data mining. *Proceedings of the Fifth International Conference on Extending Database Technology: Advances in Database Technology 1996, EDBT'06, Academic Press*, pp: 18-32.
- Park, B. and H. Kargupta, 2002. *Distributed Data Mining: Algorithms, Systems and Applications. Data Mining Handbook, USA.*
- Perlich, C., F. Provost and J. Simonoff, 2003. Tree induction vs. logistic regression: A learning-curve analysis. *J. Mach. Learn. Res.*, 4: 211-255.
- Robert Duin, P.W., 2002. The Combining classifier: To train or not to train? *Proceedings 16th International Conference on Pattern Recognition, 2002, Fac. of Appl. Sci., Delft Univ. of Technology*, pp: 765-770.
- Robert Shapire, E., 1990. The strength of weak learnability. *Mach. Learn. J.*, 5: 197-227.
- Rodriguez, J.J., L.I. Kuncheva and C.J. Alonso, 2006. Rotation forest: A new classifier ensemble method. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28: 1619-1630.
- Ross Quinlan, J., 1993. *C4.5 Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.*
- Ross Quinlan, J., 1996. Bagging, boosting and C4.5. *Proceedings of the 13th National Conference on Artificial Intelligence, 1996, AAAI 96, AAAI Press, Portland, Oregon.*, pp: 725-730.
- Sabine McConnell, M. and B. David Skillicorn, 2004. Building predictors from vertically distributed data. *Proceedings of the 2004 conference of the Centre for Advanced Studies on Collaborative research CASCON'04, 2004, ACM, USA.*, pp: 150-162.
- Shafer, J., R. Agarwal and M. Mehta, 1996. SPRINT: A scalable parallel classifier for data mining. *Proceedings of 22nd International Conference on Very Large Databases, 1996, Mumbai, India*, pp: 544-555.
- Skillicorn, D.B. and S.M. McConnell, 2008. Distributed prediction from vertically partitioned data. *J. Parallel Distributed Comput.*, 68: 16-36.
- Tsai, S.T. and Y. Chao-Tung, 2004. Decision Tree construction for data mining on grid computing. *Proceedings IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04), 2004, USA.*, pp: 441-447.
- Yu, C. and D.B. Skillicorn, 2001. *Parallelizing Boosting and Bagging. Queen's University, Kingston, Ontario, Canada.*
- Zhang, S.C., X.D. Wu and C.Q. Zhang, 2003. Multi-database mining. *IEEE Comput. Intell. Bull.*, 2: 5-13.