Combining Multiple Classifiers System (CMC System) for Mining Distributed Databases using Meta-Learning Approaches

Yousry El-Gamal Osama Badawy Ashraf Al-Jerjawy
yelgamal@aast.edu obadawy@aast.edu al_jerjawi@hotmail.com

Abstract

Data mining techniques aim to discover patterns and extract useful information from facts recorded in databases. A widely adopted approach to this objective is to apply various machine learning algorithms to compute predictive classifiers (models) of the available data. One of the main challenges in this research area is the development of techniques that scale up to large and possibly physically distributed databases. This paper investigates data mining techniques that scale up to large and physically distributed databases (Meta-Learning). It presents a suggested solution for ensemble of classifiers through a meta-learning. Specifically, it describes the architecture, design, and implementation of CMC System (Combining Multiple Classifiers System), a distributed data mining system that facilitates the sharing of information among participating data sites without the need of exchanging or dispatching remote data, and employs meta-learning approaches to combine the multiple classifiers that are learned. Furthermore, one of the most powerful learning ideas that boost any learning algorithm (Boosting) is used to increase the performance of this system. An empirical evaluation of the CMC system and its boosting on different databases over N distributed nodes is also computed by detailing a comprehensive set of experiments and results achieved.

1 Introduction

The number and size of databases and data warehouses grow at phenomenal rates. For these databases to be useful, data mining process (the process of extracting useful information from such datasets) must be performed [1, 3, 8]. The datasets may be inherently distributed but cannot be localized on one processing site to compute one classifier [5, 6, 9]. Most of the current machine learning algorithms are required all data to be resident in main memory, which is clearly untenable in many realistic databases [15]. So, analyzing and monitoring these distributed data sources require data mining technology designed for distributed applications (Distributed Data Mining (DDM)) [2, 4, 5, 6, 7, 8, 9, 10, 11, 16, 27, 28, 29, 31, 32].

Nowadays, classifier ensembles are often used for distributed data mining in order to discover knowledge from distributed data sources. They are rather recent sub-area of machine learning that has been mainly used for increasing the predictive accuracy of single classifies, scaling up learning algorithm to very large datasets, and learning from distributed datasets. They have been applied with success to a number of applications, playing an important role to new research areas like DDM, Multiple Classifier Systems, and Information Fusion.

A particular approach that has been successfully applied to classifier ensembles is meta-learning [2, 11, 12]. Meta-learning is defined as learning from learned knowledge. It refers to a general strategy that seeks to learn how to integrate a number of separate learning processes in an intelligent fashion. The basic idea is to execute a number of machine learning processes on a number of data subsets in parallel, and then to integrate their collective results (classifiers) through an additional phase of learning.

The rest of the paper is organized as follows. Section 2 views related methodologies for combining classifiers induced from distributed databases. Section 3 presents the architecture design of the CMC System. Section 4 outlines the materials and methods used in this study. Section 5 exhibits experimental results and discussion. Section 6 concludes this work, and section 7 poses future research directions.

2 Related Works

Classification is the derivation of a function or model, called a classifier, which determines the class of an object based on its attributes. A set of objects is given as the training set in which every object is represented by a vector of attributes along with its class. A classifier is constructed by analyzing the relationship between the attributes and the classes of the objects in the training set. Such a classifier can be used to predict the classes of future objects and to develop a better understanding of the classes of the objects in the database.

The integration of multiple classifiers has been under active research in machine learning and neural networks. The challenge of integration is to decide which classifier to rely on or how to combine classifications produced by several classifiers. Two main approaches have lately been used: selection of the best classifier and combining the classifications produced by the basic classifiers.

Classifier selection methods can also be divided into two subsets: static and dynamic methods. A static method proposes one “best” method for the whole data space, while a dynamic method takes into account characteristics of a new instance to be classified [26, 30].

The most popular and simplest method of combining classifiers is voting [6, 17]. The classifications produced by the basic classifiers are handled as (un-weighted (Majority) or weighted) votes for those particular classifications and the classification with most votes is selected as the final classification. More sophisticated classification algorithms that use combination of classifiers include the Stacking [18], Average Stacking [13], Distributed Stacking [27], SCANN [19], Bagging [14, 24], Boosting [22, 23, 24], and Similarity Based [28, 33].

Stacking is a method that combines multiple classifiers by learning the way that their output correlates with the true class on an independent dataset which is collected from all nodes (meta-data) and then generates a global classifier (GC). On the other hand, Average Stacking uses the average of all classifiers outputs to generate a meta-data, and Distributed Stacking constructs a global classifier from local classifiers that does not require moving raw data around, scales up efficiently with respect to large numbers of distributed databases. SCANN method is based on the correspondence analysis [20] and the nearest neighbor procedure, combining minimal nearest neighbor classifiers within the stacking framework.

Similarity Based approach for distributed classification uses the pair-wise similarity of local classifiers in order to produce a better classification for each of the distributed databases. This is achieved by averaging the decisions of all local classifiers weighted by their similarity with the classifier induced from the origin of the unclassified instances.

Boosting is one of the most powerful learning ideas introduced in the last ten years [21, 25]. It was developed as a method for boosting the performance of any learning algorithm. The focus of boosting method is to produce a series of weak classifiers in order to produce a powerful combination. The training set used for each classifier of the series is chosen based on the performance of the earlier classifier. Instances that are incorrectly predicted by previous classifier in the series are chosen more often than instances that were correctly predicted.

All these Meta-Learning approaches and their properties are briefly described in [37].

3 CMC System

One of the main objectives of this study is the design and implementation of a system that supports Combining Multiple Classifiers (CMC), efficiently and accurately, induced from large and distributed databases. With meta-learning to provide the means for combining information across separate data sources (by integrating individually computed classifiers), CMC system is developed using four meta-learning algorithms that facilitates the sharing of information among multiple sites without the need of exchanging or directly accessing remote data. Also, one of the most powerful learning ideas that boost any learning algorithm (Boosting) is used to increase the performance of this system, which helps to choose a suitable algorithm that has more advantages upon others. This section describes the distributed architecture and design of CMC system.

3.1 CMC System Architecture

This system is designed around the idea of meta-learning to benefit from its inherent parallelism and distributed nature. Recall that meta-learning improves efficiency by executing in parallel the same or different serial learning algorithms over different sets of the training databases. A graphical representation of CMC System is depicted in

Figure 1 in a simplified scenario. In this figure, m classifiers are derived from m distributed databases. Then, the meta-learning process integrates the m classifiers.

CMC System is architected as a distributed computing construct developed on top of operating system environments. It can be viewed as a parallel application, with each constituent process running on a separate database site. Under normal operation, each site (database site) functions autonomously and exchanges classifiers with the rest. It is implemented as a collection of classification programs. CMC System consists of:

- Distributed databases.
- One or more classification programs, or in other words machine learning programs that is locally stored as native programs.
- One or more meta-learning programs, programs capable of combining a collection of classifiers.
- A repository of locally computed and imported base- and meta-classifiers.
- A Text-based User Interface.

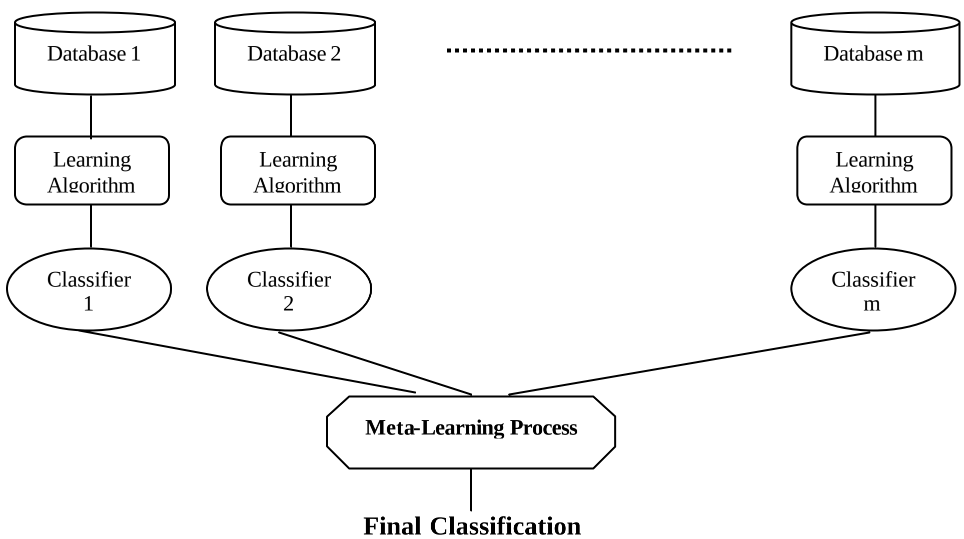

Figure 1: CMC System Architecture.

When CMC System is initiated, local learning programs are executed on the local database to compute the local classifiers. Each CMC System site may then imports (remote) classifiers and combines them with its own local classifier using the local meta-learning programs. Finally, once the base and meta-classifiers are generated, the CMC System manages the execution of these classifiers to classify new unlabeled instances. Each CMC System site stores its base- and meta-classifiers in its classifier repository (a special file for each classifier).

Top

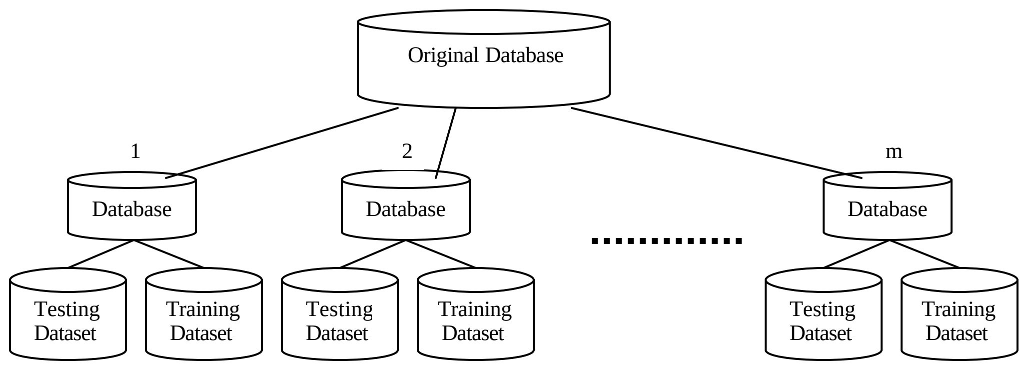

Figure 2 Simulation of distributed nodes scenario.

4 Experimental Setting

An empirical evaluation of CMC system on different databases is computed by detailing a comprehensive set of experiments. This section gives the results achieved from these experiments and offers some discussions upon them. Before discussing these various experiments and results, the used materials and methodologies are detailed.

4.1 Machine Learning Algorithm with Java Implementation

C4.5 (decision tree based algorithm [34, 35, 36]) represents a supervised approach to classification [38]. It is a simple structure where non-terminal nodes represent tests on one or more attributes and terminal nodes reflect decision outcomes. This inductive learning algorithm is used in these experiments, which is obtained from the WEKA machine-learning package [41].

4.2 Meta-Learning Algorithms with Java Implementation

Four meta-learning techniques (Majority Voting (MV), Stacking (S), Average Stacking (AS), and Distributed Stacking (DS)) are employed. These techniques are implemented using Java.

4.3 Distributed Databases

In order to evaluate the proposed framework, a set of experiments were conducted using four of the largest real world and synthetic databases from the UCI (University of California at Irvine) Machine Learning Repository [15]. The details of these databases are described in Table 1.

In this study, the original database is randomly fragmented horizontally into a variable number of databases (10, 20, 30, 40, 50, 60, 70, 80), which are divided randomly into testing dataset (20%) and training dataset (80%). This allows us to simulate a variety of distributed database configurations and examine how the methodologies scale up with respect to the number of distributed nodes. Figure 2 illustrates that.

Top

Table 1 Details of databases used in the experiments.

Data Set	Size	Attributes	Classes	Missing Values (%)
Adult	48,842	14	2	0.95
Letter	20,000	16	26	0
Nursery	12,960	8	5	0
Waveform	5,000	40	3	0

4.4 Performance Evaluation

The performance of a proposed CMC System is evaluated using the most common metrics used in the evaluation of classifier's performance (Accuracy, Error Rate, True Positive Rate, False Positive Rate, Precision, and ROC curve). All described in [8, 40, 42].

Accuracy measurements for these systems are obtained using a simple holdout method, which is usually applied. Moreover, in order to obtain realistic results, each experiment is performed 10 times then the final results are derived from averaging the partial results of each run.

The time consumed to compute the training and testing phase, which is a part of performance evaluation, is also measured. Furthermore, in order to get a machine independent measure of computational complexity, the size of the Meta-Dataset (MD) is recorded too, which is equal to (Number of records (R)* Number of attributes (A) → (KRA)). This can be helpful especially in cases where the large number of distributed nodes and classes makes the experiments too time consuming to be completed.

5 Experimental Results and Discussion

The experiments are conducted on a Personal Computer with a 2.4 GHz Pentium processor and 256 MBytes running Windows 2000 Prof.

According to the previous discussion, the main features of the experimental setting can be summarized as follow:

1. After 10 running times the average is calculated.
2. N distributed nodes (10, 20, 30, 40, 50, 60, 70, and 80) are tested.
3. Four large databases are used.
4. C4.5 decision tree learning algorithm, Boosting algorithm, and four meta-learning algorithms (Majority Voting (MV), Stacking (S), Average Stacking (AS), and Distributed Stacking (DS)) are applied.

Un-Boosted CMC System

- CMC System using Majority Voting Algorithm (CMCMV)

Top

- CMC System using Stacking Algorithm (CMCS)
- CMC System using Average Stacking Algorithm (CMCAS)
- CMC System using Distributed Stacking Algorithm (CMCDS)

Boosted CMC System

- CMC System using Boosted Majority Voting Algorithm (CMCBMV)
- CMC System using Boosted Stacking Algorithm (CMCBS)
- CMC System using Boosted Average Stacking Algorithm (CMCBAS)
- CMC System using Boosted Distributed Stacking Algorithm (CMCBDS)

5. Different performance's metrics (Accuracy, Error Rate, TP Rate, FP Rate, (TP-FP) Rate, Precision, Training Phase Time (Sec), Testing Phase Time (Sec), and Meta-Data Size (KRA)) are measured.

The following sections give the results achieved from the experiments done over Un-Boosted and Boosted CMC System using statistical tables and graphics. They also interpret why these results have occurred.

5.1 Results of Un-Boosted and Boosted CMC System

The experimental results (Accuracy, (TP-FP) Rate, Precision, Training Time (sec), Testing Time (sec), and Meta-Data size (KRA)) of Un-Boosted and Boosted CMC System are averaged over N distributed nodes and plotted in Figure 3. The full analysis of this study is available in [39].

As shown in this figure, there is no consistent pattern that can be derived from all the experiments. However, Accuracy, (TP-FP) Rate, and Precision have approximately same behavior. A first general pattern that can be noticed is that as the number of distributed nodes increases, the performance of all the strategies decreases. Moreover, CMCS has better (Accuracy, (TP-FP) Rate, and Precision) than others especially in large number of distributed nodes. Also, it is observed that CMCAS is standing in the middle of CMCS and CMCMV, while CMCDS is the worst one. This is logic as Stacking uses a large validation dataset collected from all nodes and full outputs of local classifiers' predictions (meta-dataset) to generate a Global Classifier (GC), while Average Stacking uses only the averages of the predictions according to the class, and Distributed Stacking uses these averages (meta-dataset) to learn a GC.

However, the difference in accuracy is small comparing with the difference in time. Each one of CMCMV, CMCAS, and CMCDS has a computational complexity that is independent on the number of distributed nodes and achieves tractability of building the final model in contrast to CMCS that requires a great amount

Top

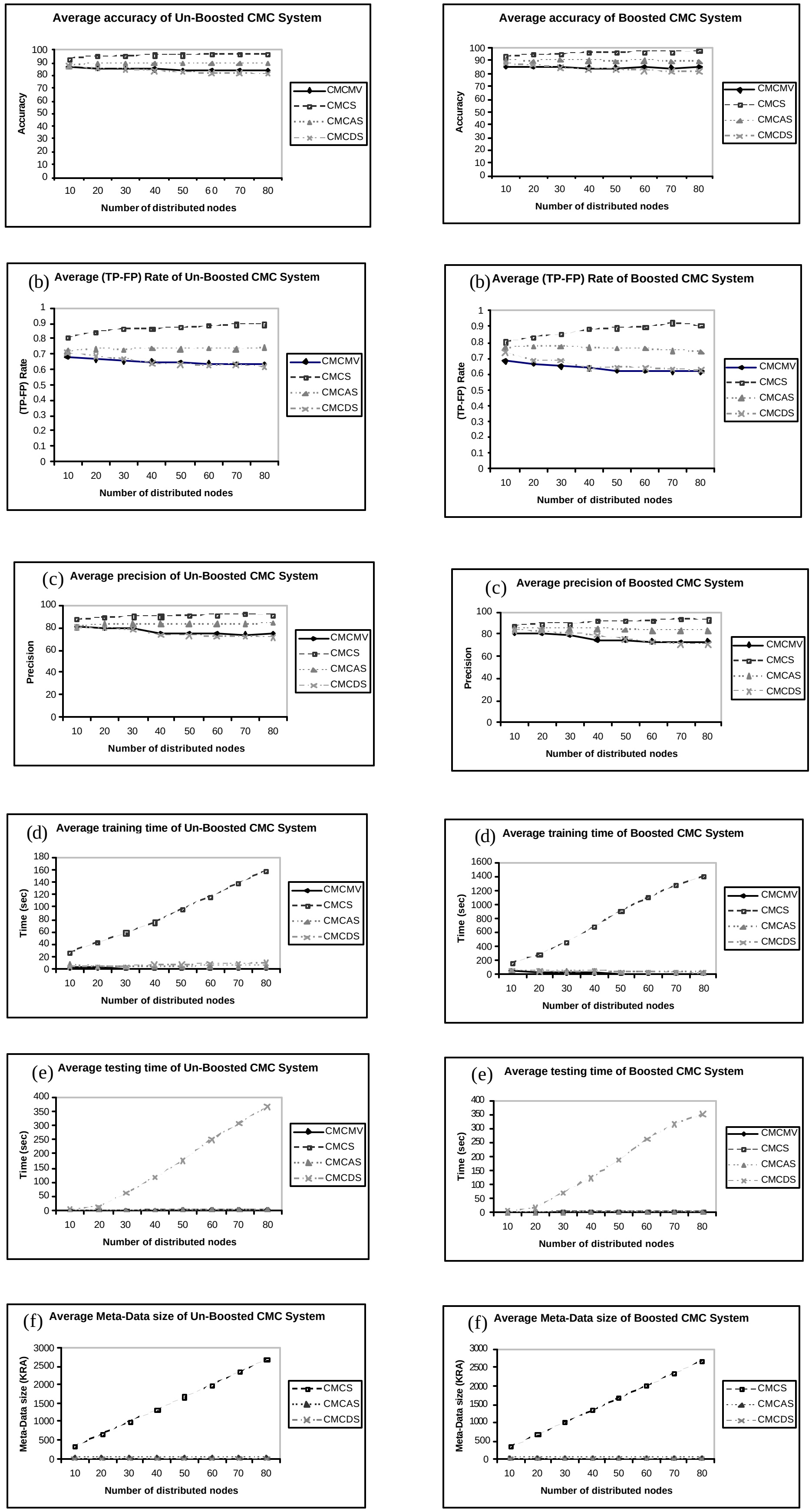

Figure 3: Average Performance of Un-Boosted and Boosted CMC System.

Top

of time for training at the training phase. This is also evident from the plots in Figure 3 (d), which depict the relationship of the necessary time to finish the training phase in the experiments among CMCMV, CMCS, CMCAS, and CMCDS. This figure shows that the training time for CMCS increases linearly with respect to the number of distributed sites. This occurs because when the full output of the local classifiers is used, the attribute vector in the meta-dataset increases by number of classes. The attribute vector in a problem with C classes and M distributed nodes has size equal to C*M. Therefore, depending on the algorithm used, the complexity of the learning problem at the training phase is related to the product C*M*N, where N is the number of records in the meta-dataset.

Also, Figure 3 (d) illustrates that the training time for CMCMV, CMCAS, and CMCDS stay in most cases almost constant with respect to the number of distributed nodes. When the averages of the local classifiers' outputs are used, the attribute vector in the meta-dataset is constant and equal to the number of classes. The size of the attribute vector in a problem with C classes and N distributed nodes will always be equal to C. Therefore, depending on the algorithm used, the complexity of the learning problem at the training phase is related to the product C*N.

So, it is concluded that each of CMCMV, CMCAS, and CMCDS has constant complexity with respect to the number of classes and linear complexity with respect to the number of distributed nodes. In contrast, CMCS has not only linear complexity with respect to the number of distributed nodes it also has one with respect to the number of classes. This is why the largest meta-data size was seen in CMCS. Figure 3 (f) illustrates that.

5.2 Un-Boosted vs. Boosted CMC System

Boosted CMC System has some advantages and disadvantages over Un-Boosted CMC System. Table 2 illustrates the overall average performance of both.

Table 2: Overall Average Performance of Un-Boosted and Boosted CMC System.

CMC System	Accuracy (%)	Error Rate	TP Rate	FP Rate	(TP-FP) Rate	Precision (%)	Training Time (Sec)	Testing Time (Sec)	Meta-Data Size (KRA)
Un-Boosted	88.21	11.79	0.796	0.072	0.724	81.26	25.46	41.36	512.35
Boosted	88.71	11.29	0.805	0.069	0.733	82.01	218.74	42.63	512.35

As can be seen from Figure 4, Boosted CMC System has better accuracy and precision than Un-Boosted CMC System. Unfortunately, as the accuracy increased the required time increased too. Boosted CMC System uses boosting algorithm that boosts the performance of learning algorithm. It has a computational complexity that is dependent on the number of iterations used for generating of boosted classifier, where in these experiments was ten. Otherwise, both have the same testing time and meta-data size.

Top

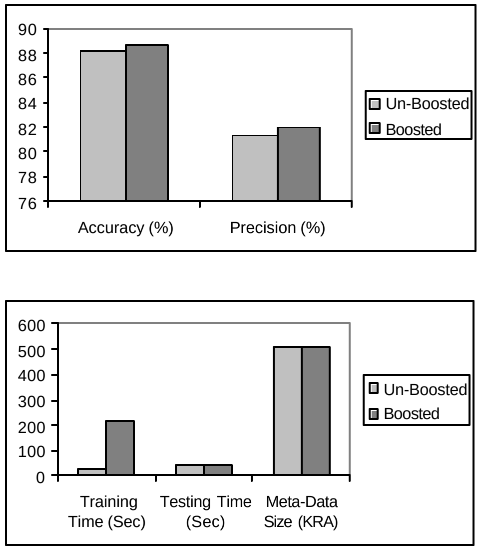

Figure 4: Overall Average Performance of Un-Boosted and Boosted CMC System.

5.3 Ensembles vs. Single Classifiers

One of the main features of meta-learning is improving predictive accuracy by combining different inductive classifiers. It is expected to derive a higher level learned model that explains a large database more accurately than individual learner. C4.5 is tested on the four databases described in Table 1 without any splitting, in order to show its performance before applying the four meta-learning algorithms. Table 3 illustrates that by presenting the overall average performance of a single classifier (C4.5) and Combining Multiple Classifiers (CMC) using the four different meta-learning algorithms.

Table 3: Overall Average Performance of Single and CMC.

Algorithm	Accuracy (%)	Error Rate (%)	Precision (%)	Training Time (Sec)	Testing Time (Sec)	Meta-Data Size (KRA)
C4.5	83.37	16.63	79.78	11.83	0.073	---
CMCMV	84.67	15.33	76.60	1.73	1.37	---
CMCS	95.5	4.5	90.29	89.49	1.55	1499.29
CMCAS	89.25	10.75	83.19	4.36	1.166	36.5
CMCDS	83.41	16.59	74.96	6.28	161.42	1.25

The superiority of combining multiple classifiers is obvious from Figure 5. Most of the combined methods perform better (Accuracy and Precision) than the individual learner. By performing different initial classifiers and combining the outputs, the final classifier may provide a better approximation to the true class. Due to the limited amount of training data, the individual classifier may not represent the true class. Thus, through

Top

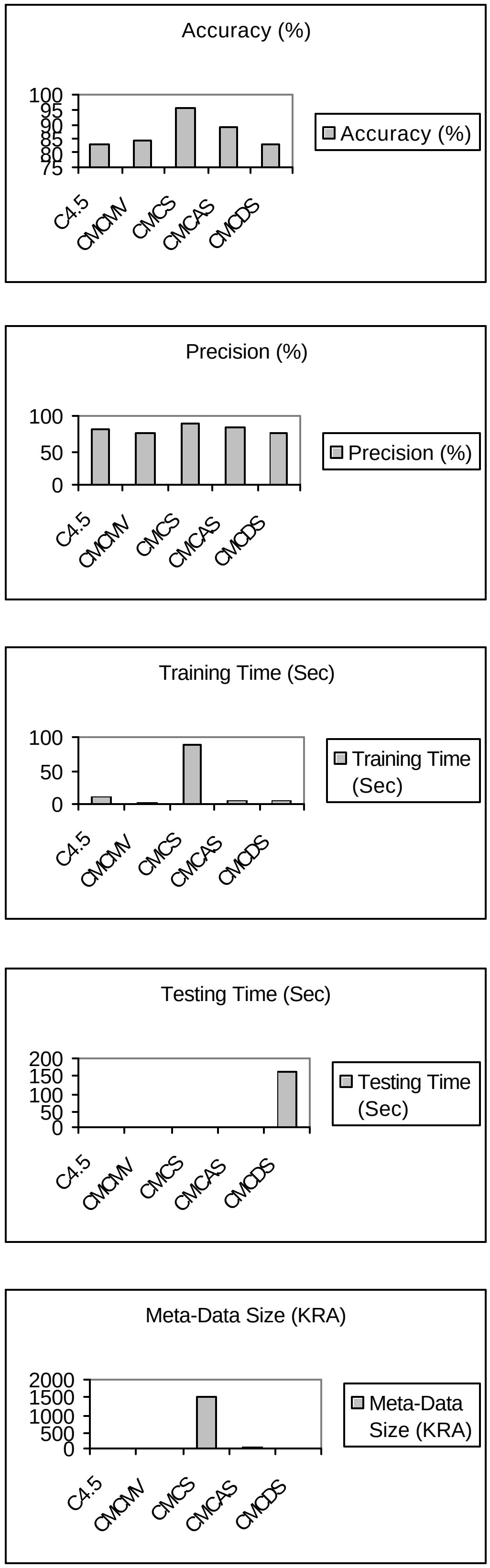

Figure 5: Overall Average Performance of Single and CMC.

Top

considering different classifiers, it may be possible to expand the final classifier to an approximate representation of the true class.

Also, it can be observed that the individual learner consumed more training time than others, except CMCS which uses the full outputs of local classifiers' predictions to generate a Global Classifier (GC) producing high accuracy up upon others. Furthermore, individual learner consumed less testing time than others because it has no meta-dataset and has one classifier to generate the final classification.

5.4 Experimental Results Summary

The combination of various learned classifiers with multiple meta-learning techniques produces many ways for mining distributed databases. Through exhaustive experiments performed, several performance measures (i.e., Accuracy, Error Rate, True Positive Rate, False Positive Rate, Precision, Training Time (sec), Testing Time (sec), and Meta-Data size (KRA)) were evaluated to determine the usefulness and effectiveness of CMC System. Its limitation was exposed too.

The empirical evaluation revealed that no specific and definitive strategy gives best results in all cases. Thus, the results produced from the experiments can be very briefly summarized by the following statements:

- CMC System could not consistently perform well over all the databases. The performance of the learning algorithms is highly dependant on the nature of the training data.
- Combining Multiple Classifiers (CMC) perform better than the individual ones in terms of their predictive accuracy and precision.
- Most of the time, the literature reports mention that a learning scheme performs better than another in term of one classifier's accuracy when applied to a particular data set. This study showed that accuracy is not the ultimate measurement when comparing the classifier's credibility. Accuracy is just the measurement of the total correctly classified instances. This measurement is the overall error rate. Thus, when comparing the performance of different classifiers, accuracy as a measure is not enough. Different measures should be evaluated depending on what type of question that the user seeks to answer (i.e, True Positive (TP), False Positive (FP), and Precision).
- Boosted CMC System has better accuracy and precision than Un-Boosted CMC System. Unfortunately, as the accuracy increased the required time increased too.
- Although CMCS has better (Accuracy, (TP-FP) Rate, and Precision) than others especially in large number of distributed nodes, it has a high computational complexity, a large meta-data size (large meta-classifier), and consumes a great amount of time for training phase.
- CMCAS exhibits a reasonable classification accuracy and precision with low computational complexity, training time, and meta-data size (meta-classifier).

Top

- CMCDS dose not require a raw data being moved around the distributed nodes as a validation dataset (low meta-data size), and it scales up well with respect to large number of classifiers, but it produces a low accuracy and precision rate.
- In general, as the number of distributed nodes increases, the performance of all the strategies decreases.

6 Conclusion

The main focus of this paper is on the Ensemble of Classifiers for Mining Distributed Databases using Meta-Learning Approaches, or in other words, on the design and implementation of a system that supports Combining Multiple Classifiers (CMC), efficiently and accurately, induced from large and distributed databases.

With meta-learning to provide the means for combining information across separate data sources (by integrating individually computed classifiers), CMC system is developed using four meta-learning algorithms (Majority Voting (MV), Stacking (S), Average Stacking (AS), and Distributed Stacking (DS)), that facilitates the sharing of information among multiple sites without the need of exchanging or directly accessing remote data. Also, one of the most powerful learning ideas that boost any learning algorithm (Boosting) is used to increase the performance of this system, which helps to choose a suitable algorithm that has more advantages upon others.

An empirical evaluation of CMC system on different databases is computed by detailing a comprehensive set of experiments. Also results achieved from these experiments and some discussions upon them are reported. The following summarizes briefly the contribution of this study:

- The adaptation of current meta-learning techniques to combine classifiers computed over data collected from different sites.
- The design of CMC System, a novel distributed data mining system that is based on meta-learning.
- The implementation of a prototype of the CMC System.
- A detailed account on the similarities and differences between several meta-learning methods. The analysis examines and contrasts the applicability of the Boosting method on these techniques for combining classifiers (MV, S, AS, and DS).
- The application of CMC System on the real-world datasets and the evaluation of its performance under different realistic metrics (Accuracy, Error Rate, True Positive Rate, False Positive Rate, Precision, Training Time (sec), Testing Time (sec), and Meta-Data Size (KRA)).
- A thorough evaluation and comparison of the performance of the CMC System and its boosting, and a deep analysis of their strengths and weaknesses.

Finally, it is believed that CMC System will be an important contributing technology to deploy mining knowledge facilities in global-scale, integrated distributed information systems.

Top

7 Future Work

This study could be further extended in the following directions:

- The implementation of CMC System is currently executed on a single machine, where the classifiers are induced sequentially from datasets. The next step is to implement a distributed computing architecture that will allow the parallel execution of the local and global model learning phases.
- CMC System should be extended to mine distributed databases with different schema (Heterogeneous).
- This research plans to explore the effectiveness of other meta-learning techniques and also intends to experiment with different learning algorithms than C4.5, which could potentially improve the performance of CMC System.

All in all, there are still many open questions and enormous opportunities to improve the suggested topic "Combining Multiple Classifiers for Mining Distributed Databases using Meta-Learning Approaches".

8 Acknowledgments

We would like to thank colleagues in the AAST for constructive discussions. We would also like to thank the contributors to the UCI collection of machine learning databases and WEKA machine-learning package.

9 References

[1] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, "From Data Mining to Knowledge Discovery: An Overview", Advances in Knowledge Discovery and Data Mining. AAAI Press,Menlo Park, CA, 1996.

[2] H. Kargupta and P. Chan (Eds.), "Advances in Distributed and Parallel Knowledge Discovery", AAAI Press, 2000.

[3] Heikki Mannila, "Data mining: machine learning, statistics, and databases", Department of Computer Science University of Helsinki. URL: http://www.cs.helsinki.fi/~mannila/.

[4] MOHAMMED J. ZAKI, YI PAN, "Introduction: Recent Developments in Parallel and Distributed Data Mining", Distributed and Parallel Databases. Kluwer Academic Publishers, 2002.

[5] Vincent Wing Sing Cho. PhD thesis, "Knowledge Discovery from Distributed and Textual Data", The Hong Kong University of Science and Technology, Hong Kong, June 1999.

Top

[6] A. Prodromidis, “Management of Intelligent Learning Agents in Distributed Data Mining Systems”. PhD thesis, Department of Computer Science, Columbia University, New York, NY,1999.

[7] A. Prodromidis, P. Chan, and S. Stolfo, “Meta-learning in distributed data mining systems: Issues and approaches”

[8] Margaret H. Dunham, “Data Mining: Introductory and Advanced Topics”, published by Prentice Hall by year 2003.

[9] B. H. Park and H. Kargupta, “Distributed Data Mining: Algorithms, Systems, and Applications”, Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, 2002.

[10] Guo, Y., and Sutiwaraphun, J., “Distributed learning with knowledge probing”, A new framework for distributed data mining. In Advances in distributed and parallel knowledge discovery, eds: H. Kargupta and P. Chan, AAAI Press, 2000.

[11] Kargupta, H. Park, B., Hershbereger, D., Johnson, E., "Collective data mining: A new perspective toward distributed data mining". Accepted in the Advances in Distributed Data Mining, Eds: Hillol Kargupta and Philip Chan, AAAI/MIT Press, 1999.

[12] P. Chan and S. Stolfo, “Meta-learning for multistrategy and parallel learning”. In Proc. Second Intl. Work. Multistrategy Learning, 150–165, 1993.

[13] Grigorios Tsoumakas and Ioannis Vlahavas. “Distributed Data Mining of Large Classifier Ensembles”. Department of Informatics, Aristotle University of Thessaloniki, Greece, 2002.

[14] Breiman, L., “Bagging predictors”, Machine Learning, 1996.

[15] Merz, C., and Murphy, P. 1998. UCI repository of machine learning databases, “http://www.ics.uci.edu/~mlearn/mlrepository.html”. Dept. of Info. and Computer Sci., Univ. of California, Irvine, CA.

[16] Prodromidis, A. L., and Stolfo, S. J., “Pruning meta-classi?ers in a distributed data mining system”. In Proc of the KDD’98 workshop in Distributed Data Mining, 1998.

[17] C. Merz, “Dynamical Selection of Learning Algorithms”. In: D. Fisher, H.-J.Lenz (Eds.), Learning from Data, Artificial Intelligence and Statistics, Springer Verlag, NY, 1996.

[8] David Wolpert, “Stacked generalization”, Neural Networks, 5, 241–259, 1992.

[19] Merz, C. “Using correspondence analysis to combine classifiers”. Machine Learning. In press, 1999.

[20] Greenacre, M. J. Theory and Application of Correspondence Analysis. London: Academic Press, 1984.

[21] Efron, B. & Tibshirani, R. “An Introduction to the Bootstrap”, 1993.

[22] Schapire. “The Strength of Weak Learn ability”. Machine Learning, Vol. 5, No. 2, 1990.

[23] Freund, Y., Schapire. ”A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting”. In: Proc. 2nd European Conf. on Computational Learning Theory, Springer-Verlag, 1995.

[24] Bauer, E., Kohavi, R. “An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants”. Machine Learning, 1999.

[25] Kohavi, R. “A study of cross-validation and bootstrap for accuracy estimation and model selection”. In C. Mellish (Ed.), the 14-th Int. Joint Conf. on Artificial Intelligence. Morgan Kaufmann, 1995.

[26] Merz, C.J. 1996. “Dynamical selection of learning algorithms”. In D. Fisher & H.-J. Lenz (Eds.) Artificial Intelligence and Statistics. New York: Springer-Verlag, 1996.

[27] Grigorios Tsoumakas and Ioannis Vlahavas. “Effective Stacking of Distributed Classifiers”. Department of Informatics, Aristotle University of Thessaloniki, Greece, 2002.

[28] Grigorios Tsoumakas, Lefteris Angelis, and Ioannis Vlahavas. “Similarity Based Distributed Classification”. Department of Informatics, Aristotle University of Thessaloniki, Greece, 2002.

[29] P. Seppo, T. Vagan, T. Alexey, “A Dynamic Integration Algorithm for an Ensemble of Classifiers”. University of Finland, P.O.Box 35, FIN-40351, Finland, 1999.

[30] Giorgio Giacinto and Fabio Roli, “Dynamic Classifier Selection”. Dept. of Electrical and Electronic Eng., University of Cagliari, Piazza d’Armi, 09123, Italy, 2001.

[31] Shichao Zhang, Xindong Wu, and Chengqi Zhang, “Multi-Database Mining”. IEEE Computational Intelligence Bulletin, 2003.

[32] Seppo Puuronen, Vagan Terziyan, and Alexander Logvinovsky, “Mining Several Databases with an Ensemble of Classifiers”. Kharkov State Technical University of Radioelectronics, Ukraine, 2003.

Top

[33] Tao Li, Shenghuo Zhu, and Mitsunori Ogihara, “A New Distributed Data Mining Model Based on Similarity”. University of Rochester, Rochester, 2003.

[34] T. Mitchell, “Decision Tree Learning”, in T. Mitchell, Machine Learning, The McGraw-Hill Companies, Inc., 1997.

[35] P. Winston, “Learning by Building Identification Trees”, in P. Winston, Artificial Intelligence, Addison-Wesley Publishing Company, 1992.

[36] H.Hamilton. E. Gurak, L. Findlater W. Olive, “http://www.cs.uregina.ca/~dbd/cs831/notes/ml/dtrees/4_dtrees1.html”.

[37] Yousry El-Gamal, Osama Badawy, and Ashraf Al-Jerjawi, “An Ensemble of Classifiers for Mining Distributed Databases using Meta-Learning Approaches”, In Proc of the International Arab Conference on Information Technology, Vol. 1, pp 262-272, Dec 2003.

[38] Quinlan J.R., “Programs for Machine Learning”. Morgan Kaufmann, San Mateo, CA, 1993.

[39] Ashraf Mohamed Al-Jerjawy, MSc thesis, “An Ensemble of Classifiers for Mining Distributed Databases using Meta-Learning Approaches”, Computer Engineering Department, Arab Academy for Science, Technology and Maritime Transport, Egypt, 2004.

[40] Jiawei Han, Micheline Kamber, “Data Mining: Concepts and Techniques”, The Morgan Kaufmann Series in Data Management Systems, 2001.

[41] Witten, I.H. and Frank, E. (No date available). “Data Mining: Practical machine learning tools with Java implementations”. Morgan Kaufmann, San Francisco. Accessed via URL: http://www.cs.waikato.ac.nz/nl/weka/, 2002.

[42] Provost, Fawcett, & Kohavi, “The case against accuracy estimation for comparing induction algorithms”. In Proc. Fifteenth Intl. Conf. Machine Learning, 1998.

Top