Distributed Data Mining:
Why Do More Than Aggregating Models

Mohamed Aoun-Allah and **Guy Mineau**
Computer Science and Software Engineering Department
Laval University, Quebec City, Canada
{Mohamed.Aoun-Allah, Guy.Mineau}@ift.ulaval.ca

Abstract

In this paper we deal with the problem of mining large distributed databases. We show that the aggregation of models, i.e., sets of disjoint classification rules, each built over a subdatabase is quite enough to get an aggregated model that is both predictive and descriptive, that presents excellent prediction capability and that is conceptually much simpler than the comparable techniques. These results are made possible by lifting the disjoint cover constraint on the aggregated model and by the use of a confidence coefficient associated with each rule in a weighted majority vote.

1 Introduction

This paper deals with the problem of mining several large and geographically distributed databases (DB_i) with the goal of producing a set of classification rules that explains the various groupings found in the observed data. The result of this mining is both a predictive and descriptive meta-classifier. In other words, we aim at producing a model which is not only capable of predicting the class of new objects, but which is also able to explain the choices of its predictions. We believe that this kind of models, based on classification rules, should also be easy to understand by humans, which is also one of our objectives. Also, our application context is one where it is impossible to gather all these databases on the same site, and this, either because of downloading time, or because of the difficulty to mine the aggregated database.

In the literature, we find very few distributed data mining techniques which are both predictive and descriptive. The majority of them try to produce a meta-classifier in the form of a set of rules with disjoint cover, i.e., where an object is covered by one and only one rule. We will show in this paper that this constraint of disjoint cover is not necessary to produce a reliable meta-classifier and furthermore, introduces unnecessary complexity to the mining technique. Therefore, we propose a simpler technique where an object can be covered by several rules, and where the lifting of this constraint enables us to produce a conceptually very simple classifier with good prediction capability as will be shown below.

The performance of our meta-classifier (from a prediction point of view) is compared to C4.5 applied on the whole database $DB = \cup_i DB_i$; it is used only as a reference to assess the potential loss of accuracy of our method since, by assumption, we stated that we could not process DB because of download/processing time constraints.

This paper proceeds as follows. We present in sections 2 a survey of some well known model aggregation techniques. Then, in section 3, we present our solution for distributed data mining (DDM) by model aggregation (DDM-MA) based on a majority vote that is pondered by some confidence coefficient. Section 4 introduces a conceptual comparison between our method and those found in the literature. In section 5, we present experimental results which prove the viability of our method. We will show that it bares comparable accuracy rates while being simpler that other DDM methods. We finally present a conclusion and our future work.

2 Model aggregation existing techniques

As we present in this paper a technique developed in a *distributed data mining* perspective, we will ignore some non relevant techniques as the *Ruler System* [Fayyad *et al.*, 1993] [Fayyad *et al.*, 1996] that was developed for the aggregation of several decision trees build on the same data set in a centralized system, the *Distributed Learning System* [Sikora and Shaw, 1996] developed in a context of information management system that builds a distributed learning system, and the *Fragmentation Approach* [Wüthrich, 1995] which uses probalistic rules. Also, we will ignore purely predictive techniques such as *bagging* [Breiman, 1996], *boosting* [Schapire, 1990], *stacking* [Tsoumakas and Vlahavas, 2002], and the *arbiter* and *combiner* methods [Chan, 1996], [Prodromidis *et al.*, 2000].

2.1 The MIL algorithm

The MIL algorithm (*Multiple Induction Learning*) was initially proposed by Williams [Williams, 1990] in order to resolve conflicts between conflictual rules in expert systems. Authors of [Hall *et al.*, 1998a; 1998b] took again the technique of Williams [Williams, 1990] to aggregate decision trees built in parallel and transformed beforehand into rules. The process of aggregation proposed by these authors is a regrouping of rules accompanied by a process of resolution of the possible conflicts between them. It should be noted that this resolution of the conflicts treats only one pair of conflict rules at a time. Two rules are considered in conflict when their premises are consistent while they produce two different classes [Williams, 1990] (called conflict of type I) , or when the conditions of the premises overlap partially [Hall *et al.*, 1998a] (called conflict of type II) or when the rules have the same number of predicates with different values for conditions and they classify objects into the same class [Hall *et al.*, 1998b] (called conflict of type III).The conflict resolution consists in either specializing one or the two rules in conflict (conflicts type I and II), or in adjusting the value of the condition, i.e., the test boundary, for the conflicts of type II and III and eventually in combining the two rules in conflict (conflict of type III). In certain cases (conflicts of type I and II), new rules are added based on the training sets to recover the cover lost by the specialization process.

2.2 The DRL system (Distributed Rule Learner)

The DRL technique (*Distributed Rule Learner*) [Provost and Hennessy, 1996] was conceived based on the advantage of the invariant-partitioning property [Provost and Hennessy, 1994]. The DRL technique begins by partitioning the training data E into nd disjoined subsets, assigns each one (E_i) to a different machine, and provides the infrastructure for the communication between different learners (named RL). When a rule r satisfies the evaluation criterion for a subset of the data (i.e., $f'(r, E_i, nd) \geq c$; f' being an evaluation function[1] of a rule and c a constant), it becomes a candidate to satisfy the global evaluation criterion; the extended invariant-partitioning property guarantees that each rule which is satisfactory on the whole data set will be acceptable at least on one subset. When a local learner discovers an acceptable rule, it sends the rule to the other machines so that they update its statistics on the remainder of the examples. If the rule meets the global evaluation criterion ($f(r, E) \geq c$; f being the principal evaluation function and c a constant), it is asserted as a satisfactory rule. In the opposite case, its local statistics are replaced by the global statistics and the rule is made available to be specialized some more. The property of invariant-partitioning guarantees that each satisfactory rule on the whole data set will be found by at least one of the RLs.

2.3 Combining rule sets generated in parallel

The work presented by [Hall *et al.*, 1999] is a mixture of the last two techniques presented above, i.e., that of [Williams, 1990], [Hall *et al.*, 1998b] and [Provost and Hennessy, 1996]. In details, they associate to each rule a measurement of its ”quality” which is based on its prediction precision as well as on the number and the type of the examples that it covers.

The technique suggested in [Hall *et al.*, 1999] is based on the use of what [Provost and Hennessy, 1996] proposes (see §2.2), with a small difference where the deletion of the rule from the space of rules under consideration is made only when the rule classifies all the data of the various distributed bases, which is the case when its measure $f(r, E)$ is lower than a certain threshold. It should be noted that each rule does not ”travel” alone from one site to another, but is indeed accompanied by the values necessary to calculate the measure associated with each rule.

However, in [Hall *et al.*, 1999], the authors show that in the extreme case the property of invariant-partitioning could not be satisfied. Thus, they prove that the precision of the aggregate rule set can be very different from the precision of the rules built on the training set. Moreover, the authors show that conflicts between rules can be solved, as described by [Hall *et al.*, 1998b] and [Williams, 1990].

In addition, [Hall *et al.*, 1999] proposes a new type of conflict between rules: a rule whose premise contains some interval that overlaps an interval that is contained in the premise of a second rule. In this case, a more general rule is created by combining the two conflicting rules and by adjusting the border values of these intervals.

3 The proposed model aggregation technique

The proposed technique is very simple. We build in parallel over each distributed DB_i a model, i.e., a set of classification rules R_i, called *base classifier*. Figure 1 shows an example of such rules.

```
IF adoption_of_the_budget_resolution = n
IF physician_fee_freeze = y
THEN CLASS: republican

IF adoption_of_the_budget_resolution = u
IF physician_fee_freeze = y
THEN CLASS: democrat
```

Figure 1: An example of rules contained in a base classifier.

Then, we compute for each rule a confidence coefficient (see below for details). Finally, in a centralized site, base classifiers are aggregated in the same set of rules ($R = \cup_i R_i$) which represents our final model, called *meta-classifier*. The global algorithm of our distributed data mining technique is described by Figure 2.

```
1. Do in parallel over each database $DB_i$
   (a) Apply on $DB_i$ a classification
       algorithm producing a set of disjoint
       cover rules.  The produced set is
                  $R_i = \{r_{ik} \mid k \in [1..n_i]\}$
       where $n_i$ is the number of rules;
   (b) Compute for each $r_{ik}$ a confidence
       coefficient $c_{r_{ik}}$ (see hereafter);
2. In a central site create:
                  $R = \bigcup_{i=1...nd} R_i$
   where $nd$ is the number of distributed
   databases.
```

Figure 2: Algorithm of the proposed DDM technique.

Since different rule sets are going to be merged together

[1] This rule evaluation function could be for example the Laplace precision estimator [Segal and Etzioni, 1994] [Webb, 1995].

whereas they are issued from different data subsets, and since each rule r has its proper error rate E_r and coverage n, we compute for each rule a confidence coefficient c_r. This confidence coefficient is computed in straightforward manner from the lower bound of an error rate confidence interval proposed by [Langford, 2005]. We defined it as *one minus the worst error rate in* $(1-\delta)$ *of the time* :

$$c_r = 1 - \overline{Bin}_-(n, nE_r, \delta)$$

where $\overline{Bin}_-(n,k,\delta) \overset{def}{=} min\{r : 1 - Bin(n,k,r) \geq \delta\}$ and $Bin(n,k,r) \overset{def}{=} \sum_{i=0}^{k} \binom{n}{i} r^i (1-r)^{n-i}$

Since R is not a set of *disjoint cover* rules where an object is covered by a unique rule, we explain hereafter how we can use this meta-classifier as a predictive and descriptive model.

3.1 The use of R as a predictive model

The set R represents the aggregation of all base classifiers ($R = \cup_i R_i$). This rule set is used as a predictive model as well as a descriptive one. From a predictive point of view, the predicted class of a new object is the class predicted by a majority vote of all the rules that cover it, where the rules are weighted by their confidence coefficients[2]. It should be noted that, contrarily to what is identified in the literature (see §2), we have restricted the notion of *rules in conflict* to be those that cover the same object but with different classification results. If several rules cover the same object and predict the same class, we do not consider them as being in conflict.

It is to be noted that any object can be covered by at most nd rules, knowing that nd is the number of sites.

3.2 The use of R as a descriptive model

As a classification system is often developed as support to decision-making, the different rules covering an object may be proposed to the user who could then judge, from his expertise, of their relevance, helped by a confidence coefficient. Presenting to a decision maker more than one rule may have its advantages since it may provide a larger and more complete view of the "limits" of each class. We bring to mind, that in machine learning, the limit which defines separation between various classes is generally not unique nor clear cut, and consequently, several rules producing the same class can represent the "hyper-planes" separating the various classes, providing various views on these data.

4 A conceptual comparison

4.1 The MIL technique

The MIL technique [Hall *et al.*, 1998a] [Hall *et al.*, 1998b] suffers from several problems. First of all, the process of conflict resolution only specializes the rules based on the classification rules data sets. The generated rules could show poor classification ability when they are applied to new objects, especially in the case of very noisy training data. In addition, the adaptation of the technique of Williams [Williams, 1990] in order to treat distributed bases implies an increase in the volume of data exchanged between the various sites. Indeed, on the one hand, each rule travels accompanied by the index of the covered objects and, on the other hand, in the event of conflict, all the objects covered by one of the two rules in conflict must be downloaded from the training site to the site resolving the conflict.

4.2 The DRL system

The most significant disadvantage of the DRL system [Provost and Hennessy, 1996] is its execution time. Indeed, when a rule is considered to be acceptable by a given site, it must go across to all the other sites. In other words, any acceptable rule on a site must classify all the data of all the other sites. Thus, the rule must, on the one hand,"travel" through all the sites, and on the other hand, classify the data of each site. If a rule is not considered to be satisfactory on the whole data set, this rule is specialized and the process starts again if it is considered to be locally acceptable. It is clear that this process could be very time consuming.

4.3 Combining rule sets generated in parallel

As for the system of combining rule sets generated in parallel, it is identical to the previous one with a little difference: any rule generated in a given site must cross over to all the other sites. Thus, the number of rules traveling between the various sites is more significant than the number of rules of the DRL system. Consequently, it is clear that this technique is slower than the preceding one.

4.4 The proposed technique

To overcome the problems of MIL technique, the proposed one is based on a majority vote that is known to be rather a robust model against noisy data. Indeed, the prediction process gives good results especially in noisy bases (see §5 below).

In the proposed technique (see §3) rules "travel" only in one way, from the distributed database site to the central site and the amount of data is almost minimal where a rule is augmented by no more than its confidence coefficient. Thus the problem of excess communication found in the DRL system and its successor is avoided.

From an execution point of view, an asymptotic analysis was conducted in [Aounallah, 2006] and [Aounallah and Mineau, 2006] of our technique and those presented in §2 of this paper. This asymptotic analysis shows clearly that in the worst case our technique is faster than existing ones.

In the best case, we expect our technique to be at least comparable to existing ones. Since having no conflicts between different base classifiers is very rare, we believe that our technique is faster than existing ones because our technique does not conduct any conflict resolution step.

Moreover, the proposed technique is no more than a simple aggregation of base classifiers. Consequently, there is no doubt that it is conceptually by far simpler that existing comparable ones, i.e., it should be faster and simpler to implement than those found in the literature (see §2).

[2] However, in an unlikely tie situation, we propose to carry out a simple majority vote. In very rare cases, when the simple majority vote also leads to a tie, we choose the majority class in the different training sets.

5 An empirical comparison

To evaluate the performance of our DDM technique, we conducted some experiments in order to assess its prediction (accuracy) rate. We compared it to a C4.5 algorithm built on the whole data set, i.e., on the aggregation of the distributed databases. This C4.5, produces a rule set R', which is used as a reference for its accuracy rate since we assumed in the introduction that it is impossible to gather all these bases onto a single site, and this, either because of downloading time, or because of the difficulty to learn from the aggregated base because of its size. The rule set R' is considered to be the ideal case, where theoretically it is not possible to perform better than a model built on the whole data set.

The conducted experiments have been tested on nine data sets: chess end-game (King+Rook versus King+Pawn), Crx, house-votes-84, ionosphere, mushroom, pima-indians-diabetes, tic-tac-toe, Wisconsin Breast Cancer (BCW)[Mangasarian and Wolberg, 1990] and Wisconsin Diagnostic Breast Cancer (WDBC), taken from the UCI repository [Blake and Merz, 1998]. The size of these data sets varies from 351 objects to 5936 objects (Objects with missing values have been deleted). Furthermore, in order to get more realistic data sets, we introduced noise in the nine aforementioned databases, and this by reversing the class attribute[3] of successively 10%, 20%, 25% and 30% of objects. Hence, since for each data set we have, in addition to the original set, 4 other noisy sets, giving a total number of databases of 45.

In order to simulate a distributed environment, the data sets have been divided as follows. We divided each database into a test set with proportion of $1/4$. This data subset was used as a test set for our meta-classifier and for R', our reference classifier. The remaining data subset (of proportion $3/4$), was divided randomly into 2, 3, 4 or 5 data subsets in order to simulate distributed databases. The size of these bases was chosen to be disparate and in such a way so there was a significant difference between the smallest and the biggest data subset. As an example of such subdivision see Figure 3.

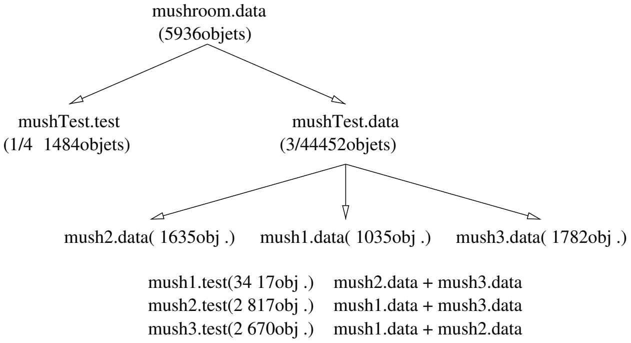

Figure 3: Example of subdivision for a database from the UCI.

Table 1: Comparison between R and R' (Original data sets).

	R'	Lower B.	Upper B.	R	Cmp.
BCW	7.0%	3.2%	10.8%	6.5%	
Chess	0.9%	0.2%	1.6%	2.5%	-
Crx	18.4%	12.5%	24.3%	19.6%	
Iono	20.5%	12.1%	28.9%	19.3%	
Mush	0.4%	0.1%	0.7%	0.3%	
Pima	23.4%	17.4%	29.4%	26.6%	
Tic-tac-toe	18.3%	13.4%	23.2%	21.3%	
Vote	3.0%	0.1%	5.9%	3.0%	
Wdbc	6.3%	2.3%	10.3%	4.9%	

Table 2: Comparison between R and R' (10% noise).

	R'	Lower B.	Upper B.	R	Cmp.
BCW	8.2%	4.1%	12.3%	5.3%	
Chess	0.9%	0.2%	1.6%	3.4%	-
Crx	14.7%	9.3%	20.1%	20.2%	-
Iono	25.0%	16.0%	34.0%	18.2%	
Mush	0.3%	0.0%	0.6%	0.3%	
Pima	33.3%	26.6%	40.0%	26.0%	+
Tic-tac-toe	20.8%	15.7%	25.9%	22.1%	
Vote	3.7%	0.5%	6.9%	3.0%	
Wdbc	7.7%	3.3%	12.1%	8.5%	

For the construction of the base classifiers we used C4.5 release 8 [Quinlan, 1996] [Quinlan, downloaded in 2004] which produces a decision tree that is then directly transformed into a set of rules. The confidence coefficient of each rule was computed using the program offered by Langford [Langford, downloaded in 2006] with a basis of 95% confidence interval (i.e., $\delta = 0.05$).

In order to assess the prediction capability of our technique we compared its prediction rate to the one of R' over the 45 aforementioned data sets. Table 1 to 5 detail the results obtained. The third and the forth columns of these tables contain respectively the lower and the upper bound of R' error rate confidence interval computed at 95% confidence. The last column contains:

- "+" if our technique outperforms R',
- "-" if R' outperforms our meta-classier and
- a blank if the two techniques are statistically comparable.

From these tables, we can see that our meta-classier performance is very comparable to the one of R' since in 35 cases

Table 3: Comparison between R and R' (20% noise).

	R'	Lower B.	Upper B.	R	Cmp.
BCW	8.8%	4.6%	13.0%	6.5%	
Chess	2.3%	1.3%	3.3%	2.8%	
Crx	21.5%	15.2%	27.8%	15.3%	
Iono	38.6%	28.4%	48.8%	34.1%	
Mush	0.3%	0.0%	0.6%	0.5%	
Pima	28.1%	21.7%	34.5%	28.1%	
Tic-tac-toe	18.8%	13.9%	23.7%	20.4%	
Vote	8.1%	3.5%	12.7%	2.2%	+
Wdbc	12.7%	7.2%	18.2%	14.8%	

[3] Please note that all data sets have a binary class attribute.

Table 4: Comparison between R and R' (25% noise).

	R'	Lower B.	Upper B.	R	Cmp.
BCW	7.1%	3.3%	10.9%	7.6%	
Chess	4.3%	2.9%	5.7%	4.8%	
Crx	23.3%	16.8%	29.8%	25.8%	
Iono	40.9%	30.6%	51.2%	36.4%	
Mush	0.3%	0.0%	0.6%	0.5%	
Pima	38.5%	31.6%	45.4%	37.0%	
Tic-tac-toe	20.8%	15.7%	25.9%	24.6%	
Vote	8.1%	3.5%	12.7%	3.0%	+
Wdbc	9.9%	5.0%	14.8%	14.1%	

Table 5: Comparison between R and R' (30% noise).

	R'	Lower B.	Upper B.	R	Cmp.
BCW	10.0%	5.5%	14.5%	7.1%	
Chess	8.1%	6.2%	10.0%	4.9%	+
Crx	33.7%	26.4%	41.0%	24.5%	+
Iono	40.9%	30.6%	51.2%	46.6%	
Mush	1.8%	1.1%	2.5%	0.5%	+
Pima	36.5%	29.7%	43.3%	36.5%	
Tic-tac-toe	24.6%	19.2%	30.0%	25.0%	
Vote	10.4%	5.3%	15.5%	4.4%	+
Wdbc	12.7%	7.2%	18.2%	16.9%	

over 45 its error rate is statistically comparable and only in 3 cases it is worst than R'. Moreover, surprisingly, our meta-classifier could outperform R' in 7 cases; this is especially the case when noise in distributed data sets is important. In these cases, we could easily see the advantage of using a noise robust model as the weighted majority vote in a non disjoint cover rule set, instead of using a single model with disjoint cover rule set.

These results, prove also the viability of the confidence coefficient proposed in this paper.

6 Conclusion

The objective of this paper is to present a very simple distributed data mining technique (DDM) by model aggregation (MA). With this intention, we presented, on the one hand, a rapid survey of existing model aggregation techniques which are most comparable to ours. And on the other hand, we presented a description of our DDM-MA technique.

Throughout this paper, we have shown that the proposed DDM technique is conceptually by far simpler that existing comparable techniques. Indeed, it consists of a simple aggregation of distributed models (base classifiers).

Experiments demonstrate that our technique, from a prediction point of view, performs as well as or even better, than a classifier built over the whole data set, R', the theoretically ideal case since it is built over the whole data. Our meta-classifier R could outperform R' due to the weighted majority vote pondered by a confidence coefficient associated to each rule. This confidence coefficient is based on the lower bound of an error rate confidence interval proposed by [Langford, 2005]. In other words, such a mojority vote over imperfect rules gives very good predictive results because the confidence coefficient used in the process uses these rules with a weight that reflects their individual prediction power.

Moreover, since the granularity of the majority vote is at rule level (instead of classifier level), the meta-classifier R can be used as a descriptive model, where the predictive class of an object is described by the rules covering it.

Due to these good results, we can imagine that our technique could be applied on very large centralized databases that could be divided into smaller ones before applying our technique, rather than applying a data mining tool over the centralized database; this is what we propose to explore in a near future.

Furthermore, we propose, on the one hand, to test the proposed technique on n-ary databases and, on the other hand, to compare it experimently to exiting techniques.

References

[Aounallah and Mineau, 2006] Mohamed Aounallah and Guy Mineau. Le forage distribué des données : une méthode simple, rapide et efficace. *Revue des Nouvelles Technologies de l'Information, extraction et gestion des connaissances*, 1(RNTI-E-6):95–106, 2006.

[Aounallah, 2006] Mohamed Aounallah. *Le forage distribué des données : une approche basée sur l'agrégation et le raffinement de modèles*. PhD thesis, Laval University, February 2006.

[Blake and Merz, 1998] C.L. Blake and C.J. Merz. UCI repository of machine learning databases. http://www.ics.uci.edu/∼mlearn/MLRepository.html, 1998.

[Breiman, 1996] Leo Breiman. Bagging predictors. *Machine Learning*, 1996.

[Chan, 1996] Philip Kin-Wah Chan. *An extensible meta-learning approach for scalable and accurate inductive learning*. PhD thesis, Columbia University, 1996.

[Fayyad *et al.*, 1993] U.M. Fayyad, N. Weir, and S. Djorgovski. Skicat: A machine learning system for automated cataloging of large scale sky surveys. In *Machine Learning: Proceedings of the Tenth International Conference*, pages 112–119, San Mateo, CA, 1993. Morgan Kaufmann.

[Fayyad *et al.*, 1996] Usama M. Fayyad, S. George Djorgovski, and Nicholas Weir. *Advances in Knowledge Discovery and Data Mining*, chapter Automating the analysis and cataloging of sky surveys, pages 471–493. AAAI Press/The MIT Press, Menlo Park, California, 1996.

[Hall *et al.*, 1998a] O. Lawrence Hall, Nitesh Chawla, and W. Kevin Bowyer. Combining Decision Trees Learned in Parallel. In *Working notes of KDD*, 1998.

[Hall *et al.*, 1998b] O. Lawrence Hall, Nitesh Chawla, and W. Kevin Bowyer. Decision tree learning on very large data sets. In *IEEE International Conference on Systems, Man, and Cybernetics, 1998*, volume 3, pages 2579–2584, oct 1998.

[Hall *et al.*, 1999] O. Lawrence Hall, Nitesh Chawla, and W. Kevin Bowyer. Learning rules from distributed data. In *Workshop on Large-Scale Parallel KDD Systems (KDD99). Also in RPI, CS Dep. Tech. Report 99-8*, pages 77–83, 1999.

[Langford, 2005] John Langford. Tutorial on practical prediction theory for classification. *Journal of Machine Learning Research*, (6):273–306, March 2005.

[Langford, downloaded in 2006] John Langford. True error bound calculation source code. http://hunch.net/∼jl/projects/prediction_bounds/bound/bound.tar.gz, downloaded in 2006.

[Mangasarian and Wolberg, 1990] O. L. Mangasarian and W. H. Wolberg. Cancer diagnosis via linear programming. *SIAM News*, 23(5):1–18, September 1990.

[Prodromidis *et al.*, 2000] Andreas L. Prodromidis, Philip K. Chan, and Salvatore J. Stolfo. Meta-learning in distributed data mining systems: Issues and approaches. In Hillol Kargupta and Philip Chan, editors, *Advances in Distributed and Parallel Knowledge Discovery*, pages 81–113. AAAI Press / MIT Press, Menlo Park, CA / Cambridge, MA, 2000. chap. 3 part II.

[Provost and Hennessy, 1994] Foster J. Provost and Daniel N. Hennessy. Distributed machine learning: Scaling up with coarse-grained parallelism. In *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, pages 340–347, 1994.

[Provost and Hennessy, 1996] Foster J. Provost and Daniel N. Hennessy. Scaling up: Distributed machine learning with cooperation. In *Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pages 74–79, 1996.

[Quinlan, 1996] J. Ross Quinlan. Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research*, 4:77–90, 1996.

[Quinlan, downloaded in 2004] J. Ross Quinlan. Source code of C4.5 release 8 algorithm. http://www.cse.unsw.edu.au/∼quinlan/c4.5r8.tar.gz, downloaded in 2004.

[Schapire, 1990] Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[Segal and Etzioni, 1994] Richard Segal and Oren Etzioni. Learning decision lists using homogeneous rules. In *National Conference on Artificial Intelligence*, pages 619–625. AAAI Press, 1994.

[Sikora and Shaw, 1996] Riyaz Sikora and Michael Shaw. A Computational Study of Distributed Rule Learning. *Information Systems Research*, 7(2):189–197, June 1996.

[Tsoumakas and Vlahavas, 2002] Grigoris Tsoumakas and Ioannis Vlahavas. Distributed Data Mining of Large Classifier Eensembles. In I.P. Vlahavas and C.D. Spyropoulos, editors, *Proceedings Companion Volume of the Second Hellenic Conference on Artificial Intelligence*, pages 249–256, Thessaloniki, Greece, April 2002.

[Webb, 1995] Geoffrey I. Webb. OPUS: An efficient admissible algorithm for unordered search. *Journal of Artificial Intelligence Research*, 3:431–465, 1995.

[Williams, 1990] Graham John Williams. *Inducing and Combining Decision Structures for Expert Systems*. PhD thesis, The Australian National University, January 1990.

[Wüthrich, 1995] Beat Wüthrich. Probabilistic knowledge bases. *IEEE Transactions on Knowledge and Data Engineering*, 7(5):691–698, 1995.